PB88-221965

User's Guide for Executing OZIPM-4 (Ozone Isopleth Plotting with Optional Mechanisms Version 4) with CBM-IV (Carbon-Bond Mechanism-IV) or Optional Mechanisms. Volume 2 Computer Code

Systems Applications, Inc., San Rafael, CA

Prepared for

Environmental Protection Agency
Research Triangle Park, NC

May 88

U.S. Department of Commerce
National Technical Information Service
NTIS

TECHNICAL REPORT DATA

(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA/SW/MT-88/053b
EPA/600/8-88/073b

3. RECIPIENT'S ACCESSION NO.
PB88-221965

4. TITLE AND SUBTITLE
USER'S GUIDE FOR EXECUTING OZIPM-4 WITH CBM-IV OR OPTIONAL MECHANISMS, VOL 2: Computer Code

5. REPORT DATE
May 1988

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)
H. Hogo and M. W. Gery

8. PERFORMING ORGANIZATION REPORT NO.

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Systems Applications, Inc.
101 Lucas Valley Road
San Rafael, California 94903

10. PROGRAM ELEMENT NO.
A101/B/63/20/5013 (FY-88)

11. CONTRACT/GRANT NO.
68-02-4136

12. SPONSORING AGENCY NAME AND ADDRESS
Atmospheric Sciences Research Laboratory-RTP, NC
Office of Research and Development
U.S. Environmental Protection Agency
Research Triangle Park, North Carolina 27711

13. TYPE OF REPORT AND PERIOD COVERED
Final (7/85 - 6/87)

14. SPONSORING AGENCY CODE
EPA/600/09

15. SUPPLEMENTARY NOTES
For magnetic tape, see: PB88-221940

16. ABSTRACT

An updated version of the Ozone Isopleth Plotting Program has been developed for use with EKMA to calculate emission reductions needed to achieve the air quality standard for ozone. The program, called OZIPM-IV (Ozone Isopleth Plotting with Optional Mechanisms - Version IV), contains the new Carbon-Bond Mechanism-IV (CBM-IV). In addition to containing a new mechanism, the OZIPM-IV expands on earlier versions of the code by providing a wider compatibility between computers, an improved isopleth plotting package, expanded user interaction with control strategy calculations and a new option for determining credits for reductions in carbon monoxide emissions. Volume 1 of the report serves as the User's Manual for OZIPM-IV. Volume 2 contains the listing of the FORTRAN computer code.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group

18. DISTRIBUTION STATEMENT
RELEASE TO PUBLIC

19. SECURITY CLASS *(This Report)*
UNCLASSIFIED

20. SECURITY CLASS *(This page)*
UNCLASSIFIED

21. NO. OF PAGES
173

22. PRICE
A08 19.95

EPA Form 2220-1 (9-73)

PB88-221965

EPA/600/8-88/073b
May 1988

USER'S GUIDE FOR EXECUTING OZIPM-4 WITH CBM-IV OR OPTIONAL MECHANISMS

Volume 2

Computer Code

by

H. Hogo
M. W. Gery

Systems Applications, Inc.
101 Lucas Valley Road
San Rafael, CA 94903

Contract No. 68-02-4136

Project Officer

Marcia C. Dodge

Atmospheric Chemistry and Physics Division
Atmospheric Sciences Research Laboratory
Research Triangle Park, NC 27711

ATMOSPHERIC SCIENCES RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
RESEARCH TRIANGLE PARK, NC 27711

REPRODUCED BY
U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
SPRINGFIELD, VA. 22161

NOTICE

The information in this document has been funded by the United States Environmental Protection Agency under Contract No. 68-02-4136 to Systems Application, Inc. It has been subjected to the Agency's peer and administrative review, and it has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ABSTRACT

The U.S. Environmental Protection Agency is continuing to recommend the use of the Empirical Kinetic Modeling Approach (EKMA) as a procedure for relating levels of photochemical ozone to levels of volatile organic compounds and oxides of nitrogen. In a recent study Systems Applications developed a new chemical mechanism called the Carbon-Bond Mechanism-IV (CBM-IV), which provides a complete and verified description of urban smog chemistry. In this study, the CBM-IV has been incorporated into a new computer program called Ozone Isopleth Plotting with Optional Mechanisms--Version 4 (OZIPM-4). This program is designed to be used with EKMA to calculate the emission reductions needed to achieve the air quality standard for ozone. The OZIPM-4 program can accept as input chemical mechanisms other than the CBM-IV. Recent updates and improvements to the OZIPM-4 program are discussed herein, along with the CBM-IV mechanism. OZIPM-4 expands on earlier codes by providing a wider compatibility between computers, an improved isopleth plotting package, expanded user interaction with control strategy calculations, and a new option for determining credits for carbon monoxide emissions reductions. Volume 1 of this report serves as the User's Manual for OZIPM-4. It contains both a detailed description of OZIPM-4 and a discussion of the input and output requirements and options available for exercising either the default CBM-IV mechanism or an optional mechanism. Five examples of OZIPM-4 input and output files are also included. Volume 2 contains the listing of the FORTRAN computer code. A magnetic tape or floppy disk of this code is also available from Systems Applications.

OZIPM-4 SOURCE LISTING

The OZIPM-4 computer program contains 54 FORTRAN routines compiled under either FORTRAN-66 or FORTRAN-77 ANSI conventions. The following table lists the routine names and the page number of each routine. Each routine in the computer code is given an alphabetical character(s) to readily identify the routine for diagnostic purposes. Further revisions can easily be implemented using the alphabetical identifiers.

Routine Name		Identifier	Page Number
Program	MAIN	A	3
Block Data	DATA	B	18
Subroutine	HDWRIT	C	23
Subroutine	MECH	D	27
Subroutine	MCHSET	E	28
Subroutine	MATRX	F	31
Subroutine	MCHWRT	G	34
Subroutine	MIXST	H	36
Subroutine	RATES	I	39
Subroutine	SPARS	J	41
Subroutine	CONVT	K	43
Subroutine	VALU	L	44
Subroutine	EMISS	M	46
Subroutine	STAIR	N	52
Subroutine	PHOT	O	55
Subroutine	SOLAR	P	57
Subroutine	SPLNA	Q	59
Subroutine	SPLNB	R	61
Function	CLOCK	S	62
Subroutine	SUNTIM	T	63
Subroutine	EKCALC	U	65
Subroutine	SIM	V	76
Subroutine	DRIVES	W	84
Subroutine	STIFFS	X	88
Subroutine	DIFFUN	Y	94
Subroutine	COSET	Z	100

Routine Name		Identifier	Page Number
Function	CLKMIN	AA	101
Subroutine	INTERP	AB	102
Subroutine	NSBSLV	AC	103
Subroutine	NSCORA	AD	104
Subroutine	NSCORD	AE	105
Subroutine	NSNFAC	AF	106
Subroutine	NSSFAC	AG	108
Subroutine	OZMX	AH	113
Subroutine	SAVIT	AI	115
Subroutine	SORDER	AJ	116
Subroutine	YSMER	AK	119
Function	CIRC	AL	120
Subroutine	SPLOT	AM	121
Subroutine	CURV1	AN	124
Function	CURV2	AO	128
Subroutine	RLINE	AP	130
Subroutine	LINER	AQ	131
Subroutine	EDGMX	AR	134
Subroutine	ISOPLT	AS	135
Subroutine	VVMNMX	AT	139
Subroutine	CONMAP	AU	140
Subroutine	VVCMAP	AV	144
Subroutine	VVKURV	AW	148
Subroutine	VVLABL	AX	153
Subroutine	VVKRV1	AY	157
Subroutine	VVKRV2	AZ	162
Subroutine	AXES	BA	165
Subroutine	FRAME	BB	168

```
C     PROGRAM MAIN
C
C***********************************************************************
C
C
C                         OZIPM-VERSION 4.00
C
C       THE OZONE ISOPLETH PLOTTING PACKAGE WITH OPTIONAL CHEMICAL
C    MECHANISM (VERSION 4.00) IS USED IN THE EMPIRICAL KINETICS
C    MODELING APPROACH (EKMA) TO ESTIMATE VOC CONTROL REQUIREMENTS
C
C       THIS VERSION CONTAINS THE CARBON-BOND CHEMICAL MECHANISM
C     (CBM-4) DESCRIBED BY GERY ET AL. (1988)
C       (EPA-600/X-XX-XXX)
C
C
C       OZIPM-4 IS AN CONDENSED VERSION OF OZIPM-3.  SUPERFICIALLY,
C     THERE ARE NO CHANGES.  THE DEFAULT MECHANISM IS THE CARBON-BOND
C     (VERSION 4) INSTEAD OF THE CBM-X IN OZIPM-3.  THE USER IS REFERRED
C     TO HOGO AND GERY (1988) FOR A DESCRIPTION OF MODEL USAGE.
C
C       OZIPM-4 IS BASED ON THE ORIGINAL OZIPP PROGRAM BY WHITTEN
C     AND HOGO (1978) AND THE OZIPM-2 PROGRAM (GIPSON, 1984).
C     THIS VERSION USES THE GEAR INTEGRATION ROUTINES WRITTEN BY
C     SPELLMANN AND HINDMARSH (1975).
C
C       PARTS OF THIS PROGRAM ARE CONTRIBUTIONS FROM JEFFRIES (1982) AND
C     GIPSON (1984).  THEORETICAL PHOTOLYSIS ROUTINE ARE FROM SCHERE AND
C     DEMERJIAN (1977).  THE SPLINE ROUTINES ARE FROM CLINE (1974).
C
C
C
C            CODE WRITTEN BY:
C
C                  H. HOGO
C                  M. W. GERY
C                  R. G. JOHNSON
C                  G. W. LUNDBERG
C
C                  SYSTEMS APPLICATIONS, INC
C                  101 LUCAS VALLEY RD
C                  SAN RAFAEL, CA 94903
C                  (415) 472-4011
C
C
C    ***************  CODE REVISION DATE:  DECEMBER, 1987
C
C                                    BY:  H. HOGO/SAI
C
C    ******** DEFAULT CHEMICAL MECHANISM:  CARBON-BOND VERSION 4
```

```
C                                    (AUGUST, 1987)
C
C                                BY:  M. GERY/SAI
C
C***********************************************************************
C
      SAVE
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /OPTS/ ISPD,ICAL,IACR,IDIL,IPLC,IEMS,IRDY,IBLK,IBCK,
     1IPLT,ITRN,ITIT,IRCT,IRAT,ITIM,IMCH,INIT,ISTR,IMIX,ISPC,ITMP
     2,IALF,IDEP,IZNI,IMAS,IMOL,ICRE
      COMMON /SCRAT3/ WY(20),ISPN
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),KKK(24),JSTRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      COMMON /ZENITH/ IPZ,ZDEF(10,20),IPHZ(20),IZENP
      COMMON /EMIS/ NEM,ISP,ESTRT(5),ESTOP,ESLP,EMSP(5),IEMLS(5),EOSLP(5
     1),EMO(26,5),ECI(5),EM(26),EC(125),ECO(125,5)
      COMMON /MIX/ NMIX,AMIX(26),STRM,STOPM,DC(104)
      COMMON /MIXING/ DSTRT,DEND,AMC(5),BMC(5),CMC(5),FD(6),FG(6),
     1 AMXX(26),DL,TTMAX,SSRISE,SRMIN,DELH,TDIL,NMXX,HEIGHT,SSET,SRISE
      COMMON /TEMPER/ TEMEND,NTEMP,QM(30)
      COMMON /FRPLOT/ SAVCON(80,5),SAVTIM(80),NTSV,INOW
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /TITL/ ITTL(36)
      COMMON /NEED/ HC,XN,NL,OZP(20),OZN(11,11,5),MR,LS,HCS,XNS
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,IIO3,
     1IIH2O,IBZA,JPLUS
      COMMON /STORE/ AST(60)
      COMMON /HOUR/ OZM(5),NGO,TTM,TM(5)
      COMMON /PLTVEC/ HCT(20),OT(20),NT,OHC,HCG,PLTGRD,OXN,XNG,HC1,XN1,T
     1ICZ,DIGZ,CHRZ
      COMMON /ALOFT/ IALFT,CALFT(10),LOCALF(10),ISPAL(10)
      COMMON /VVLB1/ FCTR,DIST,CHRSIZ,NNCHR,OZBL
      COMMON /BK1/ FBK(10),FBKAL(5),HCBK,XNBK,OZBK,H2OBK
      COMMON /CRED/ ICR,ISPCR,ISPNCR(3),SPCR69(3),SURFCR(3),ALOFCR(3),
     1REDCR(3),FSRFCR(3),FALFCR(3),COSFBK,COAFBK
      DIMENSION F(21), X(7), IPLACE(6)
      DIMENSION NRTC(135), XRTC(135),OZAR(4)
      INTEGER SPECIS,HCSPEC,PLSP,PSPEC,EMSP,JTTL(36)
```

```
C
C  INITIALIZE SOME LOCAL VARIABLES
C
      DATA HCCON/595./,XNXCON/1890./,F(21)/0.0/,ITEST/'ZO'/
      DATA IPLACE/' LOS',' ANG','ELES',', CA','LIF.','    '/
      DATA IEM,KALCMP,JS,KS1/4*0/,NC/11/
      DATA JTTL /'  ',' I','SO','PL','ET','H ',30*'  '/
C
C     CALL DATA INITIALIZATION ROUTINES
C
      WRITE (IOUT,830)
      CALL MECH
      CALL PHOT
      CALL MIXST
      CALL SUNTIM
C
C  READ INPUT FILE AND PRINT IT
C
C **** NOTE THE FOLLOWING CODE IS NOT NEEDED IN ORDER TO USE
C       OZIPM-4.00.  IF THE USER FINDS THAT THE INPUT FILE (DEFAULT
C       UNIT 5) CANNOT BE REWOUND USING  REWIND  STATEMENT THEN
C       EITHER THE INPUT UNIT NUMBER NEEDS TO BE CHANGED OR
C       THE FOLLOWING SECTION OF CODE CAN BE COMMENTED OUT.
C
      WRITE (IOUT,980)
      DO 30 I=1,1000
      READ (IN,850) ITTL
      DO 10 IJ=1,20
      IF (ITTL(IJ).NE.IBLK) GO TO 20
   10 CONTINUE
      GO TO 40
   20 IF(MOD(I,50).EQ.0) WRITE (IOUT,980)
      WRITE (IOUT,860) ITTL
   30 CONTINUE
   40 CONTINUE
      REWIND IN
C
C OPTIONS LOOP
C
      DO 820 IOPTLP=1,1000
      READ(IN,840) IOPT,(X(I),I=1,6)
C
C  DO ISOPLETH
C
      IF (IOPT.NE.ISPD) GO TO 60
      IF (ABS(X(1)).NE.0.) HC=X(1)
      IF (ABS(X(2)).NE.0.) XN=X(2)
      IF (ABS(X(4)).NE.0.) INFO=-1
      IF (ABS(X(3)).NE.0.) NL=IFIX(X(3)+0.1)
      IF (ABS(X(3)).NE.0.) READ(IN,880) (OZP(I),I=1,NL)
      IF (ABS(X(5)).NE.O.) FCTR=X(5)
      IF (ABS(X(6)).NE.O.) NUMPSP=IFIX(X(6)+0.1)
      CALL HDWRIT(IPLACE,IEM)
C
C  PERFORM DIAGRAM CALCULATIONS
C
      CALL RLINE (NC,F,KS1,JS)
      DO 50 K=1,NOZ
      IF (NUMPSP.NE.O.AND.K.GT.1) READ (IN,880) (OZP(I),I=1,NL)
      ISPN=PLSP(K)
      CALL LINER (SIG,KALCMP,F,OZN(1,1,K))
   50 CONTINUE
      INFO=0
      JS=0
      GO TO 820
C
C  DO INDIVIDUAL CALCULATIONS
C
   60 IF (IOPT.NE.ICAL) GO TO 80
      NPTO=1
      IF (ABS(X(3)).NE.O.) INFO=1
      TPRNT=60.
      TSTEP=60.
      IF (ABS(X(4)).NE.O.) TPRNT=X(4)
      IF (ABS(X(5)).NE.O.) TSTEP=X(5)
      NTSV=0
      CALL HDWRIT(IPLACE,IEM)
      CALL SIM (X(1),X(2),ZN,1)
      IF (INFO.EQ.1) WRITE (IOUT,870)
      DO 70 I=1,NOZ
      IF (TM(I).GT.O.) TMX=CLOCK(FLOAT(JSTRT),IFIX(TM(I)))
      IF (TM(I).GT.O.) WRITE (IOUT,900) PLSP(I),OZM(I),TMX
      IF (TM(I).LE.O.) WRITE (IOUT,940) PLSP(I),OZM(I)
   70 CONTINUE
      CALL SPLOT
      NPTO=0
      INFO=0
      GO TO 820
C
C  IMPLEMENT ACCURACY OPTION
C
   80 IF (IOPT.NE.IACR) GO TO 90
      IF (ABS(X(1)).NE.O.) NC=IFIX(X(1)+0.1)
      IF ((NC/2).EQ.((NC+1)/2)) NC=NC+1
      IF (ABS(X(2)).NE.O.) MR=IFIX(X(2)+0.1)
      IF (ABS(X(3)).NE.O.) ERR=X(3)
      IF (ABS(X(4)).NE.O.) SIG=X(4)
      IF (ABS(X(5)).NE.O.) SIGMA=X(5)
      IF (ABS(X(6)).NE.O.) NGO=0
      GO TO 820
```

```
C
C  READ PLACE OPTION
C
   90 IF (IOPT.NE.IPLC) GO TO 100
      IF (ABS(X(1)).NE.0.) SLA=X(1)
      IF (ABS(X(2)).NE.0.) SLO=X(2)
      IF (ABS(X(3)).NE.0.) TZ=X(3)
      IF (ABS(X(4)).NE.0.) IY=IFIX(X(4)+0.01)
      IF (ABS(X(5)).NE.0.) IM=IFIX(X(5)+0.01)
      IF (ABS(X(6)).NE.0.) ID=IFIX(X(6)+0.01)
      IF(ABS(X(1)).NE.0..OR.ABS(X(2)).NE.0.)READ(IN,930)(IPLACE(I),I=1,6
     1)
      CALL PHOT
      CALL MIXST
      CALL SUNTIM
      GO TO 820
C
C  READ MORNING AND AFTERNOON MIXING HEIGHTS
C
  100 IF (IOPT.NE.IDIL) GO TO 110
      IF (ABS(X(1)).NE.0.) Z1=X(1)
      IF (ABS(X(2)).NE.0.) Z2=X(2)
      IF (ABS(X(3)).NE.0.) DSTRT=X(3)
      IF (ABS(X(4)).NE.0.) DEND=X(4)
      AMIX(1)=-3.
      CALL MIXST
      CALL SUNTIM
      JMIN=IFIX(DSTRT)-((IFIX(DSTRT)/100)*100)
      TMIN1=FLOAT((IFIX(DSTRT)/100)*60+JMIN)
      JMIN=IFIX(DEND)-((IFIX(DEND)/100)*100)
      TMIN2=FLOAT((IFIX(DEND)/100)*60+JMIN)
      JMIN=JSTRT-((JSTRT/100)*100)
      TMIN3=FLOAT((JSTRT/100)*60+JMIN)
      TMIN4=AMAX1(TMIN1,TMIN3,SSRISE)
      TSRT=TMIN4-TMIN3
      IF (TSRT.GT.0.) DTIM=TMIN2-TMIN4
      IF (TSRT.LE.0.) DTIM=TMIN2-TMIN3
      IF (DTIM.LT.0.) DTIM=0.
      DCON=ABS(X(5))
      GO TO 820
C
C  READ HOURLY MIXING HEIGHTS
C
  110 IF (IOPT.NE.IMIX) GO TO 150
      NMIX=IFIX(X(1)+0.1)+1
      DO 120 I=1,5
  120 AMIX(I)=X(I+1)
      IF (NMIX-5.GT.0) READ(IN,880) (AMIX(I),I=6,NMIX)
C
      STRM=0.
      STOPM=FLOAT(NMIX-1)*60.
      NMX4=(NMIX-1)*3
      DO 130 I=1,NMX4
  130 DC(I)=0.
      NMIX1=NMIX-1
      DO 140 I=1,NMIX1
      K=3*I-2
      DC(K)=(AMIX(I+1)-AMIX(I))/60.
  140 CONTINUE
      DTIM=STOPM
      TSRT=0.
      GO TO 820
C
C  READ HOURLY TEMPERATURE DATA
C
  150 IF (IOPT.NE.ITMP) GO TO 170
      NTEMP=IFIX(X(1)+0.1)+1
      DO 160 I=1,5
  160 QM(I)=X(I+1)
      IF (NTEMP-5.GT.0) READ(IN,880) (QM(I),I=6,NTEMP)
      QM(NTEMP+1)=2.*QM(NTEMP)-QM(NTEMP-1)
      QM(NTEMP+2)=3.*QM(NTEMP)-2.*QM(NTEMP-1)
      TEMEND=FLOAT(NTEMP-1)*60.
      TEMP=QM(1)
      GO TO 820
C
C  READ EMISSIONS
C
  170 IF (IOPT.NE.IEMS) GO TO 220
      NEM=IFIX(X(1)+0.1)
      IEM=IFIX(ABS(X(1))+0.1)
      ESTOP=FLOAT(IEM)*60.
      IF (X(1).LT.(-0.99999999)) GO TO 190
      DO 180 I=1,5
  180 EM(I)=ABS(X(I+1))
      IF (NEM-5.GT.0) READ(IN,880) (EM(I),I=6,NEM)
      CALL EMISS (NEM,EM,EC)
      ISP=2
      EMSP(1)=IIHC
      EMSP(2)=IINX
      GO TO 820
  190 DO 200 I=1,5
  200 EMO(I,1)=X(I+1)
      IF (EMO(1,1).LT.(-0.0001)) GO TO 210
      ISP=1
      EMSP(1)=IIHC
      IF (IEM-5.GT.0) READ(IN,880) (EMO(I,1),I=6,IEM)
      CALL EMISS (IEM,EMO(1,1),ECO(1,1))
  210 READ(IN,880) (EMO(I,2),I=1,7)
      IF (EMO(1,2).LT.(-0.0001)) GO TO 820
      ISP=2
      EMSP(2)=IINX
      IF (IEM-7.GT.0) READ(IN,880) (EMO(I,2),I=8,IEM)
      CALL EMISS (IEM,EMO(1,2),ECO(1,2))
      GO TO 820
C
C  READ EMISSIONS (MOLAR UNITS)
C
  220 IF (IOPT.NE.IMOL) GO TO 300
      NEM=IFIX(X(1)+0.1)
      IEM=IFIX(ABS(X(1))+0.1)
      ESTOP=FLOAT(IEM)*60.
      IF (X(1).LT.(-0.99999999)) GO TO 240
      DO 230 I=1,5
  230 EM(I)=ABS(X(I+1))
      IF (NEM-5.GT.0) READ(IN,880) (EM(I),I=6,NEM)
      CALL EMISS (NEM,EM,EC)
      GO TO 820
  240 ISP=IFIX(ABS(X(2))+0.1)
      FHT=X(3)
      DAREA=X(4)
      HCEM=X(5)
      XNEM=X(6)
      EFACT=1.
      IF (FHT.GT.0.) EFACT=24450./FHT
      IF (FHT.GT.0..AND.DAREA.GT.0.) EFACT=24450./(FHT*DAREA)
      IF (ISP.GT.2) READ (IN,880) (SPRSE(I),I=3,ISP)
      SPRSE(1)=HCEM
      SPRSE(2)=XNEM
      DO 290 I=1,ISP
      READ (IN,840) EMSP(I),(EMO(J,I),J=1,6)
      IF (IEM-6.GT.0) READ (IN,880) (EMO(J,I),J=7,IEM)
C
      IF (EMSP(I).NE.IIHC.AND.EMSP(I).NE.IINX) GO TO 260
      CONCEN=HCEM
      IF (EMSP(I).EQ.IINX) CONCEN=XNEM
      QO=EFACT/CONCEN
      DO 250 J=1,IEM
      EMO(J,I)=EMO(J,I)*QO
  250 CONTINUE
      GO TO 280
C
C  CONVERT OTHER SPECIES
C
  260 CONTINUE
      QO=EFACT/SPRSE(I)
      DO 270 J=1,IEM
      EMO(J,I)=EMO(J,I)*QO
  270 CONTINUE
  280 CALL EMISS (IEM,EMO(1,I),ECO(1,I))
  290 CONTINUE
      GO TO 820
C
C  READ EMISSIONS (MASS UNITS)
C  THIS OPTION WAS IMPLEMENTED IN OZIPM-2
C
  300 IF (IOPT.NE.IMAS) GO TO 330
      NEM=IFIX(X(1)+0.1)
      IEM=IFIX(ABS(X(1))+0.1)
      ESTOP=FLOAT(IEM)*60.
      ISP=2
      EMSP(1)=IIHC
      EMSP(2)=IINX
      READ (IN,880) (EMO(I,1),I=1,IEM)
      READ (IN,880) (EMO(I,2),I=1,IEM)
      HCEM=X(2)
      XNEM=X(3)
      FHT=X(4)*.001
      DO 320 I=1,ISP
      CONCEN=HCEM
      IF (EMSP(I).EQ.IINX) CONCEN=XNEM
      CONV=HCCON
      IF (EMSP(I).EQ.IINX) CONV=XNXCON
      QO=1./(CONV*CONCEN*FHT)
      DO 310 J=1,IEM
      EMO(J,I)=EMO(J,I)*QO
  310 CONTINUE
      CALL EMISS (IEM,EMO(1,I),ECO(1,I))
  320 CONTINUE
      ISP=2
      GO TO 820
C
C  READ CALCOMP PLOTTING OPTION
C
  330 IF (IOPT.NE.IPLT) GO TO 340
      KALCMP=1
      FCTR=0.6
      HC1=8.5
      XN1=5.95
      CHRZ=0.10
      IF (ABS(X(1)).NE.0.) FCTR=X(1)
      PLTGRD=X(2)
      IF (ABS(X(3)).NE.0.) HC1=X(3)
      IF (ABS(X(4)).NE.0.) XN1=X(4)
      IF (ABS(X(5)).NE.0.) CHRZ=X(5)
      IF (ABS(X(6)).NE.0.) CHRSIZ=X(6)
      DIGZ=CHRZ
      TICZ=CHRSIZ
      GO TO 820
C
```

```
C  PERFORM VOC CONTROL REQUIREMENT ESTIMATIONS BASED ON
C     CITY-SPECIFIC EKMA GUIDELINES
C
  340 IF (IOPT.NE.ISTR) GO TO 350
C
C  ALL EKMA CALCULATIONS ARE DONE IN SUBROUTINE EKCALC
C
      CALL HDWRIT(IPLACE,IEM)
C
C  CALL EKMA SUBROUTINE TO CALCULATE VOC CONTROL REQUIREMENTS
C
      CALL EKCALC (X)
      GO TO 820
C
C  READ REACTIVITIES
C
  350 IF (IOPT.NE.IRCT) GO TO 360
      KHC=IFIX(X(1)+0.1)
      IF (ABS(X(2)).NE.0.) XNF(1)=X(2)
      IF (ABS(X(2)).NE.0.) XNF(2)=1.-X(2)
      IF (KHC.GT.0) READ(IN,880) (RCTY(I),I=1,KHC)
      GO TO 820
C
C  READ SPECIES OF INTEREST
C
  360 IF (IOPT.NE.ISPC) GO TO 410
      IF (ABS(X(1)).GT.0.) NOZ=IFIX(X(1)+0.1)
      READ(IN,950) (PLSP(I),I=1,NOZ)
      DO 380 J=1,NOZ
      DO 370 I=1,NS
      IF (SPECIS(I).NE.PLSP(J)) GO TO 370
      KOZ(J)=I
      GO TO 380
  370 CONTINUE
      WRITE (IOUT,970) PLSP(J)
      STOP
  380 CONTINUE
  390 IF (PLSP(1).EQ.II03 ) GO TO 820
      IF (ITTL(6).NE.ITEST ) GO TO 820
      DO 400 I=1,36
  400 ITTL(I)=JTTL(I)
      GO TO 820
C
C  READ TITLE
C
  410 IF (IOPT.NE.ITIT) GO TO 420
      READ(IN,850) (ITTL(I),I=1,36)
      GO TO 820
C
C  READ TRANSPORT OPTIONS
```

```
C
  420 IF (IOPT.NE.ITRN) GO TO 440
      OZIN=X(1)
      OZAL=X(2)
      HCIN=X(3)
      HCAL=X(4)
      XNIN=X(5)
      XNAL=X(6)
      IF (HCIN.GT.(-0.0000001)) GO TO 430
      JIN=IFIX(ABS(HCIN)+0.1)
      READ(IN,880) HCIN,(FINHC(I),I=1,JIN)
  430 IF (HCAL.GT.(-0.0000001)) GO TO 820
      JAL=IFIX(ABS(HCAL)+0.1)
      READ(IN,880) HCAL,(FALHC(I),I=1,JAL)
      GO TO 820
C
C  READ BACKGROUND CONDITIONS
C
  440 IF (IOPT.NE.IBCK) GO TO 450
      OZBK=X(1)
      HCBK=X(2)
      XNBK=X(3)
      IF (HCBK.GE.(-0.00001)) GO TO 820
      JBK=IFIX(ABS(X(2))+0.01)
      READ (IN,880) HCBK,(FBK(I),I=1,JBK)
      GO TO 820
C
C  READ ALOFT CONCENTRATIONS OF SPECIES OTHER THAN O3,NMOC,NOX
C
  450 IF (IOPT.NE.IALF) GO TO 480
      IALFT=IFIX(ABS(X(1))+0.01)
      READ (IN,950) (ISPAL(I),I=1,IALFT)
      DO 470 I=1,IALFT
      DO 460 J=1,NS
      IF (ISPAL(I).NE.SPECIS(J)) GO TO 460
      LOCALF(I)=J
      GO TO 470
  460 CONTINUE
  470 CONTINUE
      READ (IN,880) (CALFT(I),I=1,IALFT)
      GO TO 820
C
C  READ INITIAL CONDITIONS FOR ALL SPECIES OTHER THAN O3,NMOC,NOX
C
  480 IF (IOPT.NE.INIT) GO TO 490
      NI=IFIX(ABS(X(1))+0.1)
      READ(IN,950) (REACT(I),I=1,NI)
      READ(IN,880) (C(I),I=1,NI)
      GO TO 820
C
```

```
C  READ SURFACE DEPOSITION RATE (CM/SEC)
C
  490 IF (IOPT.NE.IDEP) GO TO 530
      IDPTIM=IFIX(ABS(X(1))+0.01)
      IDEPO=IFIX(ABS(X(2))+0.01)
      DPEND=FLOAT(IDPTIM)*60.
      DO 520 I=1,IDEPO
      READ (IN,840) ISPDP(I),(RDEP1(J,I),J=1,6)
      IF (IDPTIM-6.GT.0) READ (IN,880) (RDEP1(J,I),J=7,IDPTIM)
      DO 500 J=1,IDPTIM
      RDEPO(J,I)=RDEP1(J,I)*36.
  500 CONTINUE
      CALL EMISS(IDPTIM,RDEPO(1,I),RDCOEF(1,I))
      IF (ISPDP(I).EQ.IIHC) GO TO 520
      DO 510 J=1,NS
      IF (ISPDP(I).NE.SPECIS(J)) GO TO 510
      LOCDEP(I)=J
      GO TO 520
  510 CONTINUE
  520 CONTINUE
      GO TO 820
C
C  READ ALREADY COMPLETED SIMULATIONS
C
  530 IF (IOPT.NE.IRDY) GO TO 560
      KS1=IFIX(ABS(X(1))+0.1)
      JS=2
      IF (NOZ.EQ.1) WRITE (IOUT,910) PLSP(1)
      IF (NOZ.GT.1) WRITE (IOUT,910) (PLSP(I),I=1,NOZ)
      DO 550 I=1,KS1
      IF (X(1).GT.0.) READ(IN,880) HCC,XNN,ZN,OZAR
      IF (X(1).LT.(-0.98)) READ (IALL,990) HCC,XNN,ZN,OZAR
      WRITE (IALN,990) HCC,XNN,ZN,OZAR
      IF (XNN.GT.0.) RT01=HCC/XNN
      IF (XNN.LE.0.) RT01=HCC*1.0E7
      NOZM1=NOZ-1
      IF (NOZ.EQ.1) WRITE (IOUT,920) HCC,XNN,RT01,ZN
      IF (NOZ.GT.1) WRITE (IOUT,920) HCC,XNN,RT01,ZN,(OZAR(JJ),JJ=1,NO
     1ZM1)
      JJ=MOD(I,11)
      IF (JJ.EQ.0) JJ=11
      II=(I-JJ)/11 + 1
      OZN(JJ,II,1)=ZN
      IF (NOZ.LE.1) GO TO 550
      DO 540 JK=2,NOZ
      OZN(JJ,II,JK)=OZAR(JK-1)
  540 CONTINUE
  550 CONTINUE
      GO TO 820
C
```

```
C  READ CHEMICAL MECHANISM
C
  560 IF (IOPT.NE.IMCH) GO TO 650
      IZENP=1
      IF (X(1).LE.0.) CALL MCHWRT
      IF (ABS(X(5)).NE.0.) TEMP=X(5)
      IF (QM(1).GT.0.) TEMP=QM(1)
      IF (X(1).LT.0.) GO TO 820
      NX=IFIX(ABS(X(1))+0.1)
      IP=IFIX(ABS(X(2))+0.1)
      NHC=IFIX(ABS(X(3))+0.1)
      NEPA=IFIX(ABS(X(4))+0.1)
      IF (IP.NE.0) READ(IN,880) (XRTC(I),I=1,IP)
      IF (IP.EQ.0) GO TO 600
      DO 570 I=1,IP
  570 IPH(I)=IFIX(XRTC(I)+0.1)
C
C  PLACE ZENITH ANGLE DEPENDENCE INTO APPROPRIATE SLOT FOR EACH REACTION
C
      DO 590 K=1,IP
      DO 580 I=1,10
      PP(I,K)=1.
  580 CONTINUE
  590 CONTINUE
  600 IF (NHC.NE.0) READ(IN,950) (HCSPEC(I),I=1,NHC)
      IF (NHC.NE.0) READ(IN,880) (CARB(I),I=1,NHC)
      CALL MCHSET (NX,NEPA)
      DO 640 J=1,NS
      DO 610 I=1,NHC
      IF (SPECIS(J).NE.HCSPEC(I)) GO TO 610
      IH(I)=J
      GO TO 640
  610 CONTINUE
  620 IF (SPECIS(J).EQ.IINO2 ) INOX(1)=J
      IF (SPECIS(J).EQ.IINO  ) INOX(2)=J
      DO 630 K=1,NOZ
  630 IF (SPECIS(J).EQ.PLSP(K)) KOZ(K)=J
  640 CONTINUE
      INHH=1
      GO TO 820
C
C  MODIFY RATE CONSTANTS USING THE RATE OPTION
C
  650 IF (IOPT.NE.IRAT) GO TO 720
      NRR=IFIX(X(1)+.1)
      IF (ABS(X(2)).NE.0.) GO TO 700
      READ(IN,880) (XRTC(I),I=1,NRR)
      DO 660 I=1,NRR
  660 NRTC(I)=IFIX(XRTC(I)+.1)
      READ(IN,880) (XRTC(I),I=1,NRR)
      DO 690 I=1,NRR
      J=NRTC(I)
      DO 670 JK=1,IP
      IF (J.EQ.IPH(JK)) RFCT(JK)=XRTC(I)
      IF (J.EQ.IPH(JK)) GO TO 680
  670 CONTINUE
  680 CONTINUE
  690 A(J)=XRTC(I)
      GO TO 820
  700 DO 710 I=1,NR
  710 A(I)=0.
      GO TO 820
C
C  READ DESIRED SIMULATION TIME
C
  720 IF (IOPT.NE.ITIM) GO TO 730
      START=0.
      STOPP=600.
      JSTRT=800
      JSTOP=1900
      IF (ABS(X(2)).EQ.0.) GO TO 820
      JSTRT=IFIX(X(1)+0.1)
      JSTOP=IFIX(X(2)+0.1)+100
      JMIN=JSTRT-((JSTRT/100)*100)
      TMIN1=FLOAT((JSTRT/100)*60+JMIN)
      JMIN=JSTOP-((JSTOP/100)*100)
      TMIN2=FLOAT((JSTOP/100)*60+JMIN)
      STOPP=TMIN2-TMIN1
      STOPP=STOPP-60.
C
C ******* UPDATED 11/81 FOR 24 HR TIME
C
      IF (STOPP.LE.0.) STOPP=1440.+STOPP
      CALL PHOT
      CALL MIXST
      CALL SUNTIM
      GO TO 820
C
C  ZENITH ANGLE OPTION
C
  730 IF (IOPT.NE.IZNI) GO TO 780
      IPZ=IFIX(X(1)+0.1)
      DO 770 K=1,IPZ
      READ (IN,880) (XRTC(I),I=1,11)
      IPNW=IFIX(XRTC(1)+0.01)
      DO 740 J=1,IP
      IF (IPNW.EQ.IPH(J)) GO TO 750
  740 CONTINUE
      WRITE (IOUT,960) IPNW
      STOP
  750 DO 760 I=1,10
      PP(I,J)=XRTC(I+1)
      IF (J.EQ.1) RTCON(I)=RTCON(I)*XRTC(I+1)
      IF (J.EQ.1) PP(I,J)=1.
  760 CONTINUE
  770 CONTINUE
      GO TO 820
C
C  INPUT EMISSION REDUCTION CREDITS (CURRENTLY IMPLEMENTED FOR CO ONLY)
C
  780 IF (IOPT.NE.ICRE) GO TO 810
      ICR=1
      IF (X(1).GT.0.) ICR=IFIX(X(1)+0.1)
      IF (X(1).LT.-0.0001) ICR=0
      IF (X(1).LT.-0.0001) GO TO 820
      IEMCR=IFIX (ABS(X(2))+0.1)
      NEMCR=IFIX (X(2)+0.1)
      DO 800 J=1,ICR
      READ (IN,840) ISPNCR(J),SPCR69(J),SURFCR(J),ALOFCR(J),REDCR(J),
     1FSRFCR(J),FALFCR(J)
      JX=J+2
      EMSP(JX)=ISPNCR(J)
      READ (IN,880) (EMO(I,JX),I=1,7)
      IF ((IEMCR-7).GT.0) READ (IN,880) (EMO(I,JX),I=8,IEMCR)
      IF (NEMCR.GT.0) CALL EMISS (IEMCR,EMO(1,JX),ECO(1,JX))
      IF (NEMCR.GT.0) GO TO 800
      FHT=X(3)*.001
      Q0=1./(1145.*SPCR69(J)*FHT)
      DO 790 I=1,IEMCR
      EMO(I,JX)=EMO(I,JX)*Q0
  790 CONTINUE
      CALL EMISS (IEMCR,EMO(1,JX),ECO(1,JX))
  800 CONTINUE
      GO TO 820
C
C  ALL OPTIONS ARE COMPLETED, END PROGRAM
C   OR OPTION NOT FOUND
C
  810 IF (IOPT.NE.IBLK) WRITE (IOUT,890) IOPT
      IF (IOPT.EQ.IBLK) STOP
  820 CONTINUE
C
      STOP
C
C
  830 FORMAT (1H1 ///////////////////
     149X,39H*************************************** /
     249X,39H*                                     * /
     349X,39H*   OZONE ISOPLETH PLOTTING PACKAGE   * /
     449X,39H*      WITH OPTIONAL MECHANISMS       * /
     549X,39H*                                     * /                 A   701
     649X,39H*            O Z I P M                * /                 A   702
     749X,39H*                                     * /                 A   703
     849X,39H*           VERSION 4.00              * /                 A   704
     949X,39H*                                     * /                 A   705
     149X,39H*          DECEMBER, 1987             * /                 A   706
     249X,39H*                                     * /                 A   707
     349X,39H***************************************)                  A   708
  840 FORMAT (A4,6X,6F10.2)                                            A   709
  850 FORMAT (36A2)                                                    A   710
  860 FORMAT (10X,36A2)                                                A   711
  870 FORMAT (1H1//)                                                   A   712
  880 FORMAT (7F10.5)                                                  A   713
  890 FORMAT (1H1,24HTHE OPTIONS INSTRUCTION ,A4,21H CANNOT BE PROCESSEDA   714
     1.)                                                               A   715
  900 FORMAT (1H0,30X,21HMAXIMUM ONE HOUR AVE ,A4,1X,3H = ,F7.5,12H CENTA   716
     1ERED AT,F6.0,9H LDT      )                                       A   717
  910 FORMAT (1H1 // 14X,36HTHE FOLLOWING RESULTS WERE READ IN. ////1H ,A   718
     112X,2HHC,14X,3HNOX,11X,5HRATIO,13X,A4,4(10X,A4))                 A   719
  920 FORMAT (1H0,8F16.5)                                              A   720
  930 FORMAT (6A4)                                                     A   721
  940 FORMAT (1H0,30X,8HMAXIMUM ,A4,1X,44H NOT REACHED, THE LAST ONE HOUA   722
     1R AVERAGE WAS ,F7.5,5H PPM.)                                     A   723
  950 FORMAT (7(A4,6X))                                                A   724
  960 FORMAT (1H0, 9HREACTION ,I10,10H NOT FOUND)                      A   725
  970 FORMAT (1H1,39X,8HSPECIES ,A4,38H IS NOT FOUND IN THE KINETIC MECHA   726
     1ANISM)                                                           A   727
  980 FORMAT (1H1//////1H0,28HTHE INPUTS FOR THIS RUN ARE  //)         A   728
  990 FORMAT (1X,7E13.4)                                               A   729
      END                                                              A   730-
      BLOCK DATA
      SAVE
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),
     1IA(60),JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,
     2ZENI
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON/OPTS/ ISPD,ICAL,IACR,IDIL,IPLC,IEMS,IRDY,IBLK,IBCK,
     1IPLT,ITRN,ITIT,IRCT,IRAT,ITIM,IMCH,INIT,ISTR,IMIX,ISPC,ITMP
     2,IALF,IDEP,IZNI,IMAS,IMOL,ICRE
      COMMON /EMIS/ NEM,ISP,ESTRT(5),ESTOP,ESLP,EMSP(5),IEMLS(5),EOSLP(5
     1),EMO(26,5),ECI(5),EM(26),EC(125),ECO(125,5)
      COMMON /MIX/ NMIX,AMIX(26),STRM,STOPM,DC(104)
      COMMON /MIXING/ DSTRT,DEND,AMC(5),BMC(5),CMC(5),FD(6),FG(6),
     1 AMXX(26),DL,TTMAX,SRISE,SRMIN,DELH,TDIL,NMXX,HEIGHT,SSET,SSRISE
      COMMON /TEMPER/ TEMEND,NTEMP,QM(30)
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /TITL/ ITTL(36)
      COMMON /NEED/ HC,XN,NL,OZP(20),OZN(11,11,5),MR,LS,HCS,XNS
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,II03,
     1IIH2O,IBZA,JPLUS
      COMMON /HOUR/ OZM(5),NGO,TTM,TM(5)
      COMMON /PLTVEC/ HCT(20),OT(20),NT,OHC,HCG,PLTGRD,OXN,XNG,HC1,XN1,T
     1ICZ,DIGZ,CHRZ
      COMMON /VVLB1/ FCTR,DIST,CHRSIZ,NNCHR,OZBL
      COMMON /BK1/ FBK(10),FBKAL(5),HCBK,XNBK,OZBK,H2OBK
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),K(24),JSTRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /ZENITH/ IPZ,ZDEF(10,20),IPHZ(20),IZENP
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      COMMON /ALOFT/ IALFT,CALFT(10),LOCALF(10),ISPAL(10)
      COMMON /GEAR1/ TTDUM(5),UROUND,NNDUM(4)
      COMMON /PLTND/ JBAR,JSYMB,CVERT(9),TVERT(52,2)
      COMMON /EXPVAL/ EXPMAX
      COMMON /CRED/ ICR,ISPCR,ISPNCR(3),SPCR69(3),SURFCR(3),ALOFCR(3),
     1REDCR(3),FSRFCR(3),FALFCR(3),COSFBK,COAFBK
      INTEGER SPECIS,HCSPEC,PLSP,PSPEC,EMSP
C
C     PLOTTING DATA
C
      DATA JBAR/'I'/,JSYMB/'O'/
      DATA TVERT/17*'    ',' C  ',' O  ',' N  ',' C  ',' E  ',' N  ',
     1' T  ',' R  ',' A  ',' T  ',' I  ',' O  ',' N  ','    ',' P  ',
     2' P  ',' M  ',70*'    '/
C
C  UROUND (ROUND-OFF ERROR), EXPMAX IS THE MAXIMUM EXPONENT ALLOWED
C
      DATA UROUND/1.25E-7/,EXPMAX/87.4/
      DATA TEMEND/0./
      DATA ISPD/'ISOP'/,ICAL/'CALC'/,IACR/'ACCU'/,IDIL/'DILU'/,
     1     IPLC/'PLAC'/,IEMS/'EMIS'/,IRDY/'ALRE'/,IBLK/'    '/,
     2     IPLT/'PLOT'/,ITRN/'TRAN'/,ITIT/'TITL'/,IRCT/'REAC'/,
     3     IRAT/'RATE'/,ITIM/'TIME'/,IMCH/'MECH'/,INIT/'INIT'/,
     4     IMIX/'MIXI'/,ISPC/'SPEC'/,ISTR/'EKMA'/,IMOL/'MOLE'/,
     5     II03/'03  '/,IIHC/'VOC '/,IINX/'NOX '/,IBCK/'BACK'/,
     6     ITMP/'TEMP'/,IALF/'ALOF'/,IDEP/'DEPO'/,IZNI/'ZENI'/,
     7     IMAS/'MASS'/,ICRE/'CRED'/,IICO/'CO  '/,IINO2/'NO2 '/,
     8     IINO/'NO  '/,IIH2O/'H2O '/,MBLANK/'M   '/,IBLANK/'    '/,
     9     JPLUS/'+   '/
C
C  SET DEFAULTS
C
      DATA HC/2.0/,XN/0.14/,NL/11/,MR/5/
      DATA ITTL/'ST','AN','DA','RD',' 0','ZO','NE',' I','SO','PL','ET',
     1          'H ','CO','ND','IT','IO','NS',19*'  '  /
C
C  FILE I/O UNITS
C
      DATA IN/5/,IOUT/6/,IALN/9/,IALL/8/,IOZC/11/
      DATA INFO/0/,NPTO/0/,NGO/1/
      DATA OZP/.08,.12,.16,.20,.24,.28,.30,.32,.34,.36,.40,9*0./
      DATA TSRT/0.0/,DTIM/420./,Z1/510./,Z2/630./,DCON/0.0/
      DATA SIG/2./,SIGMA/2./
      DATA NMIX/0/,AMIX(1)/-3./,NEM/0/
      DATA DSTRT/800./,DEND/1500./,PLTGRD/0./,OHC/0./,FCTR/0.6/
      DATA TICZ/0.07/,DIGZ/0.10/,CHRZ/0.10/,HC1/8.5/,XN1/5.95/
      DATA CHRSIZ/0.07/,QM(1)/-3./
      DATA NOZ/1/,PLSP/'03  ',4*'    '/,
     1KOZ/4,4*0/
      DATA ICR/0/,ISPCR/0/
C
C  DEFAULT REACTIVITIES
C
      DATA RCTY/.037,.035,.052,.021,.089,.117,.564,0.,.085,11*0./,
     1    FINHC/.034,.020,.037,.070,.042,.026,.498,0.,.273,11*0./,
     2    FALHC/.034,.020,.037,.070,.042,.026,.498,0.,.273,11*0./,
     3XNF/.25,.75/,
     4FENX/.05,.95/,NI/0/,CARB/2.,2.,2.,1.,7.,8.,1.,4.,1.,11*0./
      DATA HCSPEC/'ETH ','OLE ','ALD2','FORM','TOL ','XYL ','PAR ',
     1'ISOP','NR  ',11*'    '  /
      DATA IH/25,24,17,16,26,32,21,33,34,11*0/,INOX/1,2/,NHC/9/
C
C  SET UP DEFAULT TRANSPORTED CONCENTRATIONS
```

```
C
      DATA OZIN/0./,OZAL/0./,HCIN/0./,HCAL/0./,XNIN/0./,XNAL/0./
C
C  SET UP BACKGROUND CONCENTRATIONS (HC,NOX,OZ,H2O,CO)
C
      DATA HCBK/0./,XNBK/0./,OZBK/0./,FBK/10*-3./,FBKAL/5*-3./
      DATA H2OBK/20000./,COSFBK/1.2/,COAFBK/.5/
C
C  SET DEFAULT ERROR TOLERANCE AND DEFAULT TEMPERATURE
C
      DATA ERR/0.003/,TEMP/303.0/
      DATA ESTOP/0.0/,STOPP/600./,TPRNT/60./,TSTEP/60./
      DATA SLA/34.058/,SLO/118.250/,TZ/7.0/,IY/1986/,IM/6/,ID/21/
      DATA JSTRT/0800/,JSTOP/1900/,IINC/60/
C
C  MIXING HEIGHT DATA
C
      DATA FD/0.,.07,.14,.33,.50,.70/
      DATA AMC/32.7892,10.9811,-26.3876,7.4405,3.4242/
      DATA BMC/0.0,6.8857,9.1918,-5.8492,-2.0545/
      DATA CMC/.1251,.6071,1.7325,2.3676,1.0239/
      DATA FG/0.,.02,.10,.58,.85,1.0/
C
C
C  DATA FOR GEAR
C
      DATA NR/82/,NS/35/
      DATA R/135*0./,INHH/1/
      DATA  ISC/405*1/
      DATA IP/11/,RFCT/1.,.053,1.,33.9,0.1975,.189,1.,1.,1.,8.40,8.96,
     1 9*1./
C
C  DATA FOR SUN
C
      DATA IPHZ/1,9,34,38,39,45,69,74,8,14,23,9*0/,IPZ/11/,IZENP/-1/
      DATA RTCON/0.5893,0.5851,0.5713,0.5470,0.5093,0.4537,0.3740,
     10.2578,0.1341,0.0242/
C
C  SET DEFAULT PHOTOLYTIC VALUES AS FUNCTION OF ZENITH ANGLE
C
      DATA ZDEF/10*1.
     1,0.00461,0.00448,0.00413,0.00359,0.00287,0.00208,0.001235
     2,0.00054,0.00022,0.00013
     3,0.00575,0.00573,0.00567,0.00554,0.00535,0.00508,0.00463
     4,0.00401,0.00347,0.00525
     5,0.00370,0.00366,0.00356,0.00340,0.00314,0.00280,0.002334
     6,0.001738,0.00128,0.00177
     7,0.00575,0.00573,0.00567,0.00554,0.00535,0.00508,0.00463
     8,0.00401,0.00347,0.00525
     9,0.000589,0.000578,0.000550,0.000502,0.000435,0.000355
     1,0.000257,0.000158,0.000091,0.000094
     2,0.00370,0.00366,0.00356,0.00340,0.00314,0.00280,0.002334
     3,0.001738,0.00128,0.00177
     4,0.00370,0.00366,0.00356,0.00340,0.00314,0.00280,0.002334
     5,0.001738,0.00128,0.00177,30*1.,90*0./
      DATA IPH/1,8,9,14,23,34,38,39,45,69,74,9*0/
      DATA MAXL,MAXZ,MAXJ,LAM1,INC/09,10,47,2900,100/
      DATA  Z          /0.,10.,20.,30.,40.,50.,60.,70.,78.,86./
      DATA MNLM,MXLM/2900,4500/
C
C  DATA FOR ALOFT AND DEPOSITION
C
      DATA IALFT/0/,IDEPO/0/,DPEND/0./,IDPTIM/0/,DNOWS/0./
C
C     DATA FOR CBM4.3 MECHANISM
C
      DATA SPECIS/'NO2 ','NO  ','O   ','O3  ','NO3 ','O1D ','H2O ',
     1           'OH  ','HO2 ','N2O5','HNO3','HNO2','PNA ','H2O2',
     2           'CO  ','FORM','ALD2','C2O3','XO2 ','PAN ','PAR ',
     3           'XO2N','ROR ','OLE ','ETH ','TOL ','CRES','TO2 ',
     4           'OPEN','CRO ','MGLY','XYL ','ISOP','NR  ','M   ',
     5       26*'    ' /
      DATA KR/1,3,4,3*3,1,4,4,6,6,4,4,4*5,2*10,2,2,2,12,8,12,1,8,9,
     1 9,13,8,9,9,14,8,8,5*16,4*17,2*18,20,2*18,8,21,3*23,3,8,4,5,3
     2 ,8,4,26,2*28,8,27,30,3*29,8,8,31,3,8,4,5,19,22,19,34,55*0,2,
     3 1,1,2,4,3*0,7,8,9,0,2,1,1,7,0,2,1,8,0,2*12,8,11,2,1,0,13,9,9,
     4 0,14,15,8,0,0,3,5,3,8,5,0,2,1,0,18,9,0,8,0,0,1,4*24,3*25,8,2,0
     5 ,27,5,1,0,8,4,32,31,0,4*33,2,2,19,74*0,7,11*0,7,102*0,2,4,1,2,
     6 5,1,5,3,6,3,8,9,8,1,1,2,10,11,5,1,2*12,2,1,2,11,5,8,13,9,1,
     7 2*14,8,9,3*9,15,8,11,3*18,2*16,20,18,3*16,2*19,9,0,17,16,17,
     8 19,16,19,16,9,1,27,2*30,0,18,19,17,9,19,18,9,19,16,22,1,0,0,
     9 34,53*0,3,12*0,3,0,1,0,0,1,3*0,8,0,1,0,0,1,0,1,6*0,2*15,0,9
     1 ,9,8,0,11,9,1,0,1,3*19,22,17,0,0,9,17,2*16,9,16,15,19,9,9,19,
     2 11,0,9,15,18,19,18,9,17,16,17,71*0,2,25*0,2*15,3*0,15,9,0,0,
     3 5*9,0,0,19,21,19,22,15,17,9,27,29,0,9,0,0,15,9,16,27,0,15,24,
     4 9,25,102*0,2*19,3*0,8,0,17,21,0,0,15,19,8,17,19,9,0,28,0,0,29,
     5 3*0,18,19,31,0,0,19,22,31,109*0,21,22,0,0,16,9,15,1,8,9*0,16,
     6 15,21,0,0,15,25,21,109*0,2*23,0,0,22,0,9,21,11*0,8,28,0,0,25,
     7 31,15,113*0,21,0,21,12*0,9,3*0,21,18,9,113*0,8,14*0,31,4*0,17,
     8 8,193*0/
      DATA A/1.0,4.3230E+06,26.64,13750.,2309.,2438.,.04731,.053,1.0,
     1 424600.,3.26,100.,3.,33.9,44160.,.5901,1853.,1.9E-06,2.776,
     2 1.5390E-04,1.6E-11,9799.,.1975,9770.,1.5E-05,16820.,217.9,
     3 12270.,2025.,5.115,6833.,4144.,.2181,.189,2520.,322.,15000.,
     4 1.0,1.0,237.,.93,636.,24000.,3.7,1.0,18315.,12230.,.0222,3700.,
     5 9600.,21.,1203.,137100.,95445.,22000.,5920.,42000.,.018,11.35,
     6 1080.,11920.,2.7020E-03,9150.,12000.,250.,61000.,32500.,20000.,
     7 8.40,44000.,.015,36200.,26000.,8.96,27000.,142000.,.018,470.,
     8 12000.,1000.,2000.,1.0,53*0.0/
      DATA S/0.0,-1175.,1370.,0.0,-687.,-602.,2450.,0.0,0.0,-390.,
1 0.0,940.,580.,0.0,-250.,1230.,-256.,0.0,10897.,-530.,0.0,          B    201
2 -806.,3*0.0,-713.,-1000.,-240.,-749.,10121.,-380.,-1150.,          B    202
3 -5800.,0.0,187.,4*0.0,1550.,0.0,986.,-250.,2*0.0,-250.,-5500.,     B    203
4 14000.,2*0.0,1710.,0.0,8000.,2*0.0,324.,-504.,2105.,0.0,792.,      B    204
5 -411.,2633.,-322.,7*0.0,500.,-116.,8*0.,-1300.,54*0.0/             B    205
 DATA SC/415*1.,2.,2*1.,.89,2.,2*1.,2.,1.,2*2.,12*1.,2.,3*1.,        B    206
1  2.,10*1.,2.,.79,1.,.87,.96,2*1.,.63,1.,.5,.91,3*1.,.44,.9,1.,     B    207
2  .4,4*1.,.03,.7,2*1.,.6,73*1.,.89,30*1.,2.,3*1.,2.,.79,1.,.13,     B    208
3  1.1,1.,-2.1,.38,1.,.74,1.,1.7,1.56,.42,.08,.9,1.,.6,1.,.3,1.,2.   B    209
4  ,.62,.5,2*1.,.8,1.,.4,2*1.,.2,68*1.,.11,34*1.,2.,.79,1.,.11,      B    210
5  .94,.04,.02,.28,-1.,.22,.09,1.,.22,.12,.36,.9,.56,.6,.3,2*1.,     B    211
6  2.,.7,.2,2*1.,.55,.67,.55,.1,.1,.06,.1,103*1.,.79 ,1.,.11,        B    212
7  -2.1,2*1.,.3,1.,.1,1.,.7,2*1.,.56,2*1.,.3,4*1.,.03,.8,2*1.,       B    213
8  .5,.13,.2,109*1.,-.11,.04,2*1.,.2,1.,.33,1.,.3,10*1.,.69,1.1,     B    214
9  2*1.,.5,1.,.1,109*1.,.76,.02,2*1.,.02,1.,.44,-1.,11*1.,.08,.3,    B    215
1  2*1.,.45,.4,.06,113*1.,.22,1.,-1.,12*1.,.76,3*1.,.9,.2,.44,       B    216
2  113*1.,.2,14*1.,.2,4*1.,.2,.1,193*1./                             B    217
 END                                                                 B    218-
```

```
      SUBROUTINE HDWRIT(IPLACE,IEM)
      SAVE
C
C  THIS SUBROUTINE WRITES THE HEADER PAGE FOR THE CURRENT CALCULATIONS
C
C
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),KKK(24),JSTRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      COMMON /ZENITH/ IPZ,ZDEF(10,20),IPHZ(20),IZENP
      COMMON /EMIS/ NEM,ISP,ESTRT(5),ESTOP,ESLP,EMSP(5),IEMLS(5),EOSLP(5
     1),EMO(26,5),ECI(5),EM(26),EC(125),ECO(125,5)
      COMMON /MIX/ NMIX,AMIX(26),STRM,STOPM,DC(104)
      COMMON /MIXING/ DSTRT,DEND,AMC(5),BMC(5),CMC(5),FD(6),FG(6),
     1 AMXX(26),DL,TTMAX,SSRISE,SRMIN,DELH,TDIL,NMXX,HEIGHT,SSET,SRISE
      COMMON /TEMPER/ TEMEND,NTEMP,QM(30)
      COMMON /FRPLOT/ SAVCON(80,5),SAVTIM(80),NTSV,INOW
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /TITL/ ITTL(36)
      COMMON /NEED/ HC,XN,NL,OZP(20),OZN(11,11,5),MR,LS,HCS,XNS
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,II03,
     1IIH20,IBZA,JPLUS
      COMMON /STORE/ AST(60)
      COMMON /HOUR/ OZM(5),NGO,TTM,TM(5)
      COMMON /PLTVEC/ HCT(20),OT(20),NT,OHC,HCG,PLTGRD,OXN,XNG,HC1,XN1,T
     1ICZ,DIGZ,CHRZ
      COMMON /ALOFT/ IALFT,CALFT(10),LOCALF(10),ISPAL(10)
      COMMON /VVLB1/ FCTR,DIST,CHRSIZ,NNCHR,OZBL
      COMMON /BK1/ FBK(10),FBKAL(5),HCBK,XNBK,OZBK,H2OBK
      COMMON /CRED/ ICR,ISPCR,ISPNCR(3),SPCR69(3),SURFCR(3),ALOFCR(3),
     1REDCR(3),FSRFCR(3),FALFCR(3),COSFBK,COAFBK
      DIMENSION  ITMIX(26), IZC(10),XRTC(100),IPLACE(6)
      INTEGER SPECIS,HCSPEC,PLSP,PSPEC,EMSP
C
      DATA IZC/0,10,20,30,40,50,60,70,78,86/,IFRST/1/
C
      IF (IZENP.LE.0.OR.IFRST.NE.1) GO TO 30
      IFRST=2
      WRITE (IOUT,140) IZC
      DO 20 K=1,IP
      J=IPH(K)
      I=KR(J,1)
      DO 10 LL=1,10
      IF (K.EQ.1) XRTC(LL)=RTCON(LL)*RFCT(K)*PP(LL,K)
      IF (K.NE.1) XRTC(LL)=PP(LL,K)*RFCT(K)*RTCON(LL)*RFCT(1)
   10 CONTINUE
      WRITE (IOUT,150) J,SPECIS(I),(XRTC(LL),LL=1,10)
   20 CONTINUE
   30 CONTINUE
      WRITE (IOUT,160) (ITTL(I),I=1,36)
      KSTOP=JSTOP-100
      WRITE (IOUT,170) IPLACE,SLA,SLO,TZ,IM,ID,IY,JSTRT,KSTOP
      NOON=SPECIE
      IRISE=SRISE
      ISET=SSET
      IF (INFO.EQ.(-1)) WRITE (IOUT,180) NOON,IRISE,ISET
      IF (NMIX.GT.0) GO TO 50
      DDTRT=FLOAT(JSTRT)+(TSRT/60.)*100.
      DDEND=DDTRT+(DTIM/60.)*100.
      WRITE (IOUT,190) Z1,Z2,DDTRT,DDEND
      ITMIX(1)=DDTRT
      NMX=INT(DTIM/60.+0.5)+1
      DO 40 J=2,NMX
   40 ITMIX(J)=ITMIX(J-1)+100
      WRITE (IOUT,410) (ITMIX(J),J=1,NMX)
      WRITE (IOUT,420) (AMXX(J),J=1,NMX)
      IF (DCON.GT.0.) WRITE (IOUT,200) DCON
      GO TO 60
   50 NMIX1=NMIX-1
      WRITE (IOUT,360) (I,I=1,NMIX1)
      WRITE (IOUT,370) (AMIX(I),I=1,NMIX)
   60 IF (QM(1).LT.0.) GO TO 70
      NTEMP1=NTEMP-1
      WRITE (IOUT,450) (I,I=1,NTEMP1)
      WRITE (IOUT,460) (QM(I),I=1,NTEMP)
   70 WRITE (IOUT,270)
      IF (NHC.NE.0) WRITE (IOUT,210) (HCSPEC(I),RCTY(I),I=1,NHC)
      IF (HCIN.GT.0.) WRITE (IOUT,390) (HCSPEC(I),FINHC(I),I=1,NHC)
      IF (HCAL.GT.0.) WRITE (IOUT,400) (HCSPEC(I),FALHC(I),I=1,NHC)
      IF (HCBK.GT.0.) WRITE (IOUT,430) (HCSPEC(I),FBK(I),I=1,NHC)
      WRITE (IOUT,280) XNF(1)
      IF (NHC.EQ.0) WRITE (IOUT,260)
      IF((OZIN+HCIN+XNIN) .GT. 0.0 .OR. (OZAL+HCAL+XNAL) .GT. 0.0)
     1 WRITE (IOUT,380)
      IF (OZIN+HCIN+XNIN.GT.0.) WRITE (IOUT,220) OZIN,HCIN,XNIN
      IF (NI.NE.0.AND.(OZIN+HCIN+XNIN.LE.0..AND.OZAL+HCAL+XNAL.LE.0.))
     1WRITE (IOUT,380)
      IF (NI.NE.0) WRITE (IOUT,290) (REACT(I),C(I),I=1,NI)
      IF (OZAL+HCAL+XNAL.GT.0.) WRITE (IOUT,230) OZAL,HCAL,XNAL
      IF (IALFT.GT.0) WRITE (IOUT,300) (ISPAL(I),CALFT(I),I=1,IALFT)
      IF (OZBK+HCBK+XNBK.GT.0.) WRITE (IOUT,440) OZBK,HCBK,XNBK
      IF (NEM.GT.0) WRITE (IOUT,240) (I,I=1,NEM)
      IF (NEM.GT.0) WRITE (IOUT,250) (EM(I),I=1,NEM)
      IF (NEM.GT.(-1).AND.ICR.LE.0) GO TO 110
      ISP1=ISP+ICR
      IB1=1
      IF (NEM.GT.-1) IB1=3
      DO 90 J=1,NS
      DO 80 I=IB1,ISP1
      IF (EMSP(I).EQ.IIHC  .OR.EMSP(I).EQ.IINX  ) GO TO 80
      IF (SPECIS(J).NE.EMSP(I)) GO TO 80
      IEMLS(I)=J
      GO TO 90
   80 CONTINUE
   90 CONTINUE
      WRITE (IOUT,310) (J,J=1,IEM)
      DO 100 I=IB1,ISP1
  100 WRITE (IOUT,330) EMSP(I),(EMO(J,I),J=1,IEM)
  110 CONTINUE
      IF (IDPTIM.LE.0) GO TO 130
      WRITE (IOUT,320) (J,J=1,IDPTIM)
      DO 120 I=1,IDEPO
  120 WRITE (IOUT,340) ISPDP(I),(RDEP1(J,I),J=1,IDPTIM)
  130 CONTINUE
C
      RETURN
C
C  FORMAT STATEMENTS
C
  140 FORMAT (1H1 /// 42X,48HTHE FOLLOWING PHOTOLYSIS RATE CONSTANTS ARE
     1 USED // 10X,8HREACTION,3X,7HSPECIES,43X,18HZENITH ANGLE (DEG) /
     2 12X,3HNO.,18X,I2,9I10 //)
  150 FORMAT (1H0,11X,I3,8X,A4,1X,1P10E10.2)
  160 FORMAT (1H1 // 40X,36A2)
  170 FORMAT (1H0 // 40X,40HPHOTOLYTIC RATE CONSTANTS CALCULATED FOR ///
     1,49X,6A4 /// 40X,9HLATITUDE ,F10.3 // 40X, 9HLONGITUDE ,F10.3 // 4
     20X,9HTIME ZONE,F7.1 // 40X,5HDATE ,5X,3(I4,3X) // 40X,5HTIME ,7X,I
     34,3X,2HTO,3X,I4,5X,19HLOCAL DAYLIGHT TIME)
  180 FORMAT (1H0,39X,10HSOLAR NOON,I7 / 1H0,39X,10HSUNRISE   ,I7,
     17X,10HSUNSET    ,I7)
  190 FORMAT (1H0 // 40X,39HDILUTION DETERMINED FROM THE FOLLOWING  // 4
     10X,17HINVERSION HEIGHTS,5X,7HINITIAL,F7.0,5X,5HFINAL,4X,F7.0 // 40
     2X,7HTIMING ,15X,5HSTART,F9.0,5X,4HSTOP,4X,F8.0)
  200 FORMAT (1H0,39X,55HDILUTION RATE BEFORE AND AFTER THE INVERSION CH
     1ANGE WAS // 37X,1PE12.3,9H PER MIN.)
  210 FORMAT ((1H0,39X,10HEMISSIONS ,6X,3(A4,9H FRACTION,F6.3,3X)))
  220 FORMAT (/ 40X,13HSURFACE LAYER,8X,5HOZONE,F9.3,6X,11HHYDROCARBON,F
     17.3,4X,3HNOX,F10.3,4H PPM)
  230 FORMAT (/ 40X,5HALOFT,16X,5HOZONE,F9.3,6X,11HHYDROCARBON,F7.3,4X,3
     1HNOX,F10.3,4H PPM)
  240 FORMAT (1H0 // 40X,50HCONTINUOUS EMISSIONS (EXPRESSED AS THE FRACT
     1ION OF // 40X,54HINITIAL NON-BACKGROUND CONCENTRATION EMITTED PER
     2HOUR) // 40X,4HHOUR,3X,10I6 / 47X,7I6)
  250 FORMAT (1H0,39X,8HFRACTION,10F6.3 / 48X,7F6.3)
  260 FORMAT (1H0,39X,42HTHERE ARE NO HYDROCARBONS IN THE MECHANISM)
  270 FORMAT (// 40X,10HREACTIVITY)
  280 FORMAT (1H0,39X,7HNO2/NOX,6X,F6.3)
  290 FORMAT ((1H0,39X,13HSURFACE LAYER,8X,A4,1X,F9.3,6X,A4,5X,F9.3
     1,4X,A4,F9.3,4H PPM))
  300 FORMAT ((1H0,39X,13HALOFT        ,8X,A4,1X,F9.3,6X,A4,5X,F9.3
     1,4X,A4,F9.3,4H PPM))
  310 FORMAT (1H0 // 40X,70HCONTINUOUS EMISSIONS (EXPRESSED AS FRACTION
     1OF THE INITIAL PRECURSORS)
     2     // 40X,7HSPECIES,5X,4HHOUR,5X,10I6 / 61X,10I6 / 61X,10I6)
  320 FORMAT (1H0 // 40X,58HSURFACE DEPOSITION RATES (CM/SEC)
     1                    // 40X,14H
     2     // 40X,7HSPECIES,5X,4HHOUR,5X,10I6 / 61X,10I6 / 61X,10I6)
  330 FORMAT (1H0,41X,A4,4X,8HFRACTION,4X,10F6.3 / 62X,10F6.3 /
     1     62X,10F6.3)
  340 FORMAT (1H0,41X,A4,4X,8HRATE    ,4X,10F6.3 / 62X,10F6.3 /
     1     62X,10F6.3)
  350 FORMAT (1H0 // 40X,58HCONTINUOUS EMISSIONS (EXPRESSED AS THE FRACT
     1ION OF INITIAL // 40X,50HNON-BACKGROUND NOX CONCENTRATION EMITTED
     2PER HOUR) // 45X,4HHOUR,3X,11I6 / 52X,4I6)
  360 FORMAT (1H0 // 40X,39HDILUTION DETERMINED FROM THE FOLLOWING  // 4
     10X,49HINVERSION HEIGHTS (AT THE BEGINNING OF EACH HOUR) // 40X,4HH
     2OUR,9X,1H0,9I8 / 46X,10I8 / 46X,10I8)
  370 FORMAT (1H0,39X,8HHEIGHT  ,10F8.1 / 48X,10F8.1 / 48X,10F8.1)
  380 FORMAT (1H1 // 40X,26HTRANSPORTED CONCENTRATIONS )
  390 FORMAT ((1H0,39X,13HSURFACE LAYER,3X,3(A4,9H FRACTION,F6.3,3X)))
  400 FORMAT ((1H0,39X,6HALOFT ,10X,3(A4,9H FRACTION,F6.3,3X)))
  410 FORMAT (1H0 //40X,46HMIXING HEIGHTS (AT THE BEGINNING OF EACH HOUR
     1) // 40X,4HTIME,2X,10I8/46X,10I8/46X,10I8)
  420 FORMAT (1H0,39X,6HHEIGHT,2X,10F8.1/48X,10F8.1/48X,10F8.1)
  430 FORMAT ((1H0,39X,13HBACKGROUND   ,3X,3(A4,9H FRACTION,F6.3,3X)))
  440 FORMAT (/ 40X,13HBACKGROUND   ,8X,5HOZONE,F9.3,6X,11HHYDROCARBON,F
     17.3,4X,3HNOX,F10.3,4H PPM)
  450 FORMAT (1H0 // 40X,43HTEMPERATURE (AT THE BEGINNING OF EACH HOUR)
     1  // 40X,4HHOUR,9X,1H0,9I8 / 46X,10I8 / 46X,10I8)
  460 FORMAT (1H0,39X,8HTEMP      ,10F8.1 / 48X,10F8.1 / 48X,10F8.1)
      END
      SUBROUTINE MECH
      SAVE
C
C  SET DEFAULT MECHANISM CONDITIONS
C
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      COMMON /ZENITH/ IPZ,ZDEF(10,20),IPHZ(20),IZENP
C
C  PLACE ZENITH ANGLE DEPENDENCE INTO APPROPRIATE SLOT FOR EACH REACTION
C
      DO 40 K=1,IP
      DO 20 J=1,IPZ
      IF (IPH(K).NE.IPHZ(J)) GO TO 20
      DO 10 I=1,10
      PP(I,K)=ZDEF(I,J)
   10 CONTINUE
      GO TO 40
   20 CONTINUE
      DO 30 I=1,10
      PP(I,K)=1.0
   30 CONTINUE
   40 CONTINUE
C
      DO 50 I=1,IP
      J=IPH(I)
      RFCT(I)=A(J)
   50 CONTINUE
C
      RETURN
      END
      SUBROUTINE MCHSET (NX,NEPA)
      SAVE
C
C  READ NEW MECHANISM
C
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,II03,
     1IIH2O,IBZA,JPLUS
      COMMON /STORE/ AST(60)
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /GEAR7/ IRS(135,12),DDD(380)
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      DIMENSION KRS(12),SC1(12),IPL(12)
      INTEGER SPECIS,AST
      DATA JMINUS/'-'/
      ICOUNT=0
      NR=0
      NS=1
      MORX=IBLANK
      DO 10 I=1,135
      A(I)=0.
      S(I)=0.
      R(I)=0.
      DO 10 J=1,12
      IRS(I,J)=IBLANK
      IF (I.EQ.1) KRS(J)=IBLANK
      SC(I,J)=1.
   10 KR(I,J)=0
   20 NS=NS-1
      WRITE (IOUT,200)
   30 IF (NEPA.LE.0)
     1READ (IN,150) (KRS(I),I=1,3),J,SC1(1),KRS(4),MORX,(SC1(LL-3),KRS(L
     2L),LL=5,6),RTE,ENERGY
C
C  CHECK FOR MORE PRODUCTS
C
      IF (MORX.NE.IBLANK.AND.NEPA.LE.0)
     1READ (IN,160) (SC1(II-3),KRS(II),II=7,12)
      IF (NEPA.GT.0)
     1READ (IN,170) J,(KRS(I),I=1,7),RTE,ENERGY
      IF (NEPA.GT.0) GO TO 50
      IF (MORX.NE.IBLANK) GO TO 50
      DO 40 II=7,12
      SC1(II-3)=1.
      KRS(II)=IBLANK
   40 CONTINUE
   50 DO 60 II=1,12
   60 IRS(J,II)=KRS(II)
      A(J)=RTE
      S(J)=ENERGY
      DO 70 II=4,12
   70 SC(J,II)=SC1(II-3)
   80 DO 90 I=1,12
   90 IF (ABS(SC(J,I)).EQ.0.) SC(J,I)=1.
      DO 120 K=1,12
      NL=K*5
      NF=NL-4
      DO 100 LK=NF,NL
  100 AST(LK)=IBLANK
      IPL(K)=IBLANK
      IF (IRS(J,K).EQ.IBLANK) GO TO 110
      IPL(K)=JPLUS
      IF (K.EQ.1.OR.K.EQ.4) IPL(K)=IBLANK
      IF (SC(J,K).LT.0.) IPL(K)=JMINUS
  110 KK = K
      IF (IRS(J,K).NE.IBLANK) CALL VALU (SC(J,K),KK,NF,NL)
  120 CONTINUE
      ICOUNT=ICOUNT+2
      IF (IRS(J,8).NE.IBLANK) ICOUNT=ICOUNT+1
      IF (IRS(J,12).NE.IBLANK) ICOUNT=ICOUNT+1
      IF (ICOUNT.GT.50) WRITE (IOUT,200)
      IF (ICOUNT.GT.50) ICOUNT=2
      WRITE (IOUT,180)
     1           J,IPL(1),(AST(I),I=1,5),IRS(J,1),IPL(2),(AST(I),I=6,10)
     2,IRS(J,2),IPL(3),(AST(I),I=11,15),IRS(J,3),IPL(4),(AST(I),I=16,20)
     3,IRS(J,4),IPL(5),(AST(I),I=21,25),IRS(J,5),IPL(6),(AST(I),I=26,30)
     4,IRS(J,6),IPL(7),(AST(I),I=31,35),IRS(J,7),A(J),S(J)
      IF (MORX .NE. IBLANK)
     1 WRITE (IOUT,190) IPL(8),(AST(I),I=36,40),
     2 IRS(J,8),IPL(9),(AST(I),I=41,45),IRS(J,9),
     3 IPL(10),(AST(I),I=46,50),IRS(J,10),
     4 IPL(11),(AST(I),I=51,55),IRS(J,11)
      IF (MORX.NE.IBLANK.AND.IRS(J,12).NE.IBLANK)
     1WRITE (IOUT,190) IPL(12),(AST(I),I=56,60),IRS(J,12)
      KR(J,1)=100
      IF (IP.EQ.0) GO TO 140
      DO 130 I=1,IP
      IF (J.NE.IPH(I)) GO TO 130
      RFCT(I)=A(J)
      GO TO 140
  130 CONTINUE
  140 IF (J.GT.NR) NR=J
      IF (J.NE.NX) GO TO 30
      CALL MATRX
      RETURN
C
  150 FORMAT (2(A4,2X),A4,I3,F5.1,A4,A1,1X,2(F5.0,1X,A4,2X),F10.2,1X,F7.
     12)
  160 FORMAT (6(F5.0,1X,A4,2X))
  170 FORMAT (I3,2X,7(A4,1X),2F10.2)
  180 FORMAT (/2X,I3,1X,3(A1,5A1,1X,A4,2X),1H=,1X,3(A1,5A1,1X,A4,2X),
     1A1,5A1,1X,A4,3X,1PE11.3,2X,E13.3)
  190 FORMAT (45X,2X,4(A1,5A1,1X,A4,2X))
  200 FORMAT (1H1////1H0,14H THE REACTIONS,86X,13HRATE CONSTANT,2X,15HAC
     1T. ENERGY(K)  )
      END
      SUBROUTINE MATRX
      SAVE
C
C  SET UP THE INTERNAL ARRAY FOR THE CURRENT CHEMICAL MECHANISM
C
      COMMON /CALC/ NR,IR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /GEAR1/ TTDUM(5),UROUND,NNDUM(4)
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,II03,
     1IIH2O,IBZA,JPLUS
      COMMON /STORE/ AST(60)
      COMMON /GEAR7/ KR(135,12),DDD(380)
      INTEGER SPECIS
      NOLD=NS+1
      DO 170 I=1,NR
C
C     SKIP REACTIONS ALREADY PROCESSED
C
      IF (IR(I,2).EQ.NOLD.OR.IR(I,3).EQ.NOLD) IR(I,3)=99
      IF (IR(I,2).EQ.NOLD) IR(I,2)=0
      IF (IR(I,1).EQ.NOLD) IR(I,1)=99
      IF (IR(I,1).EQ.NOLD) IR(I,3)=99
      IF (IABS(IR(I,1)).NE.100) GO TO 170
C
C     ZERO OUT REACTIONS
C
      DO 10 J=2,12
   10 IR(I,J) = 0
C
C     IF LESS THAN THREE REACTANTS, FILL FIRST SLOTS.
C
      IF (KR(I,1).NE.IBLANK) GO TO 20
      IF (KR(I,3).NE.IBLANK) KR(I,1)=KR(I,3)
      IF (KR(I,3).NE.IBLANK) SC(I,1)=SC(I,3)
      SC(I,3)=1.
      KR(I,3)=IBLANK
      IF (KR(I,1).NE.IBLANK) GO TO 20
      KR(I,1)=KR(I,2)
      KR(I,2)=IBLANK
      SC(I,1)=SC(I,2)
      SC(I,2)=1.
   20 IF (KR(I,2).NE.IBLANK) GO TO 30
      IF (KR(I,3).NE.IBLANK) SC(I,2)=SC(I,3)
      SC(I,3)=1.
      IF (KR(I,3).NE.IBLANK) KR(I,2)=KR(I,3)
      KR(I,3)=IBLANK
   30 DO 40 K=4,12
C
C     GET RID OF M AS A PRODUCT
C
      IF (KR(I,K).EQ.MBLANK) KR(I,K)=IBLANK
   40 CONTINUE
C
C     IF LESS THAN FOUR PRODUCTS, FILL FIRST SLOTS.
C
      DO 60 K=4,11
         IF (KR(I,K) .NE. IBLANK) GO TO 60
         DO 50 KK=K,11
            KKK = KK + 1
            SC(I,KK) = SC(I,KKK)
            KR(I,KK) = KR(I,KKK)
   50    CONTINUE
         KR(I,12) = IBLANK
         SC(I,12) = 1.0
   60 CONTINUE
C
   70 DO 160 J=1,12
      K=KR(I,J)
C
C     PROCESS REACTANTS HERE
C
      IF (J.GT.3) GO TO 120
C
C     ALL M DEPENDENT REACTIONS ARE TO HAVE A 99 IN THE THIRD SLOT
C
      IF (K.NE.MBLANK) GO TO 110
      GO TO (80,90,100),J
   80 KR(I,1)=KR(I,2)
      SC(I,1)=SC(I,2)
   90 KR(I,2)=KR(I,3)
      SC(I,2)=SC(I,3)
      SC(I,3)=1.
      KR(I,3)=MBLANK
  100 IR(I,3)=99
  110 K=KR(I,J)
C
C     ZERO ORDER REACTIONS HAVE 99 IN FIRST SLOT
C
      IF (KR(I,1).EQ.IBLANK) IR(I,1)=99
      IF (KR(I,1).EQ.IBLANK) K=99
C
C     ALL BLANKS ARE SET EQUAL TO ZERO
C
      IF (J.EQ.3.AND.K.EQ.MBLANK) K=99
  120 IF (K.EQ.MBLANK.OR.K.EQ.IBLANK) IR(I,J)=0
      IF (K.EQ.IBLANK.OR.K.EQ.99) GO TO 160
      IF (NS.NE.0) GO TO 130
      NS=1
      GO TO 150
  130 DO 140 L=1,NS
      IF (K.NE.SPECIS(L)) GO TO 140
C
C     SLOT SET TO SPECIES NUMBER
C
      IR(I,J)=L
      GO TO 160
  140 CONTINUE
C
C     IF NO SPECIES ARE FOUND, ADD ONE TO THE LIST
C
      IF (SPECIS(NS).NE.MBLANK) NS=NS+1
  150 SPECIS(NS)=K
      C(NS+1)=C(NS)
      C(NS)=0.
      IR(I,J)=NS
  160 CONTINUE
  170 CONTINUE
      IF (SPECIS(NS).NE.MBLANK) NS=NS+1
      SPECIS(NS)=MBLANK
      DO 180 IK=1,NR
      DO 180 MT=1,3
      J=IFIX(SC(IK,MT)+UROUND)
      ISC(IK,MT)=J
  180 IF (SC(IK,MT)-FLOAT(J).GT.4.*UROUND) ISC(IK,MT)=-1
      RETURN
      END
      SUBROUTINE MCHWRT
      SAVE
C
C  WRITE THE CURRENT MECHANISM TO OUTPUT FILE
C
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,IIO3,
     1IIH2O,IBZA,JPLUS
      COMMON /STORE/ AST(60)
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      DIMENSION KRS(12),IPL(12)
      INTEGER SPECIS,AST
      DATA JMINUS/'-'/
C
      ICOUNT=0
      WRITE (IOUT,110)
      DO 50 J=1,NR
      IR=KR(J,1)
      IF (IR.EQ.0.OR.A(J).EQ.0.) GO TO 50
      DO 10 I=1,12
      II=KR(J,I)
      IF (II.GT.0) KRS(I)=SPECIS(II)
      IF (II.LE.0) KRS(I)=IBLANK
   10 CONTINUE
C
      DO 40 K=1,12
      NL=K*5
      NF=NL-4
      DO 20 LK=NF,NL
   20 AST(LK)=IBLANK
      IPL(K)=IBLANK
      IF (KRS(K).EQ.IBLANK) GO TO 30
      IPL(K)=JPLUS
      IF (K.EQ.1.OR.K.EQ.4) IPL(K)=IBLANK
      IF (SC(J,K).LT.0.) IPL(K)=JMINUS
   30 KK = K
      IF (KRS(K).NE.IBLANK) CALL VALU (SC(J,K),KK,NF,NL)
   40 CONTINUE
      ICOUNT=ICOUNT+2
      IF (KRS(8).NE.IBLANK) ICOUNT=ICOUNT+1
      IF (KRS(12).NE.IBLANK) ICOUNT=ICOUNT+1
      IF (ICOUNT.GT.50) WRITE (IOUT,110)
      IF (ICOUNT.GT.50) ICOUNT=2
      WRITE (IOUT,90)
     1          J,IPL(1),(AST(I),I=1,5),KRS(1),IPL(2),(AST(I),I=6,10),
     2KRS(2),IPL(3),(AST(I),I=11,15),KRS(3),IPL(4),(AST(I),I=16,20),
     3KRS(4),IPL(5),(AST(I),I=21,25),KRS(5),IPL(6),(AST(I),I=26,30),
     4KRS(6),IPL(7),(AST(I),I=31,35),KRS(7),A(J),S(J)
      IF (KRS(8) .NE. IBLANK)
     1 WRITE (IOUT,100) IPL(8),(AST(I),I=36,40),
     2 KRS(8),IPL(9),(AST(I),I=41,45),KRS(9),
     3 IPL(10),(AST(I),I=46,50),KRS(10),
     4 IPL(11),(AST(I),I=51,55),KRS(11)
      IF (KRS(12).NE.IBLANK)
     1WRITE (IOUT,100) IPL(12),(AST(I),I=56,60),KRS(12)
   50 CONTINUE
      RETURN
C
   60 FORMAT (2(A4,2X),A4,I3,F5.1,A4,A1,1X,2(F5.0,1X,A4,2X),F10.2,1X,F7.
     12)
   70 FORMAT (6(F5.0,1X,A4,2X))
   80 FORMAT (I3,2X,7(A4,1X),2F10.2)
   90 FORMAT (/2X,I3,1X,3(A1,5A1,1X,A4,2X),1H=,1X,3(A1,5A1,1X,A4,2X),
     1A1,5A1,1X,A4,3X,1PE11.3,2X,E13.3)
  100 FORMAT (45X,2X,4(A1,5A1,1X,A4,2X))
  110 FORMAT (1H1////1H0,14H THE REACTIONS,86X,13HRATE CONSTANT,2X,15HAC
     1T. ENERGY(K)  )
      END
      SUBROUTINE MIXST
      SAVE
C
C  SETUP MIXING HEIGHTS BASED ON MORNING AND AFTERNOON VALUES
C
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),K(24),JSTRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /MIXING/ DSTRT,DEND,A(5),B(5),C(5),FD(6),
     1Y(6),AMIX(26),DL,TTMAX,SRISE,HMIN,DELH,TDIL,NMIX,HTT,SSET,SSRISE
      INTEGER PSPEC
C
C --- DETERMINE TIME OF SUNRISE (STANDARD TIME)
C
      T=400.
   10 CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF(D.GE.1.) GO TO 20
      T=CLOCK(T,5)
      GO TO 10
C
C --- CONVERT SUNRISE TO LOCAL DAYLIGHT TIME
C
   20 TSR=T
      IF(TSR.LT.0600.) TSR=0600.
C
C --- DETERMINE SOLAR DECLINATION AND DAY LENGTH
C
      CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,1200.,D,1)
      D=D*3.14159/180.
      ARG1=-TAN(0.67399)*TAN(D)
      DL=ACOS(ARG1)/3.14159*1440.
      IDL=DL
      TSS=CLOCK(TSR,IDL)
C
C --- COMPUTE TIME MAXIMUM HEIGHT IS REACHED
C
      TMAX=.70*DL
      ITMAX=TMAX
      TTMAX=CLOCK(TSR,ITMAX)
C
C --- CONVERT ALL 2400 HOUR TIMES TO MINUTES AFTER SUNRISE
C
      SRISE=(FLOAT(IFIX(TSR/100.))*60.)+
     1 (TSR-(FLOAT(IFIX(TSR/100.))*100.))
      DDSTRT=(FLOAT(IFIX(DSTRT/100.))*60.)+
     1 (DSTRT-(FLOAT(IFIX(DSTRT/100.))*100.))
      DDEND=(FLOAT(IFIX(DEND/100.))*60.)+
     1 (DEND-(FLOAT(IFIX(DEND/100.))*100.))
```

```
      DDSTRT=DDSTRT-SRISE
      DDEND=DDEND-SRISE
C
C --- COMPUTE HMIN AND DELTA HEIGHT
C
      FD1=DDSTRT/DL
      FD2=DDEND/DL
      DO 30 J=1,5
      IF(FD1.GE.FD(J).AND.FD1.LT.FD(J+1)) II=J
      IF(FD2.GE.FD(J).AND.FD2.LT.FD(J+1)) JJ=J
      J1=JJ
   30 CONTINUE
      IF(FD1.GE.0.0) GO TO 40
      FD1=0.0
      II=1
   40 IF(FD2.LT.0.70) GO TO 50
      FD2=.7
      JJ=6
      J1=5
   50 CONTINUE
      X=FD1-FD(II)
      XX=FD2-FD(JJ)
      FG1=((A(II)*X+B(II))*X+C(II))*X+Y(II)
      FG2=((A(J1)*XX+B(J1))*XX+C(J1))*XX+Y(JJ)
      DELH=(Z2-Z1)/(FG2-FG1)
      HMIN=Z1-FG1*DELH
      HMAX=HMIN+DELH
C
C --- COMPUTE MIXING HEIGHTS AT HOURLY INTERVALS
C
      NMIX=0
      JJSTP=JSTOP-100
      IT=JSTRT/100
      T=IT*100
      IF (DEND.GT.TTMAX) DEND=TTMAX
      DO 110 J=1,26
      IF(T.LT.DSTRT) GO TO 80
      IF(T.GE.DEND) GO TO 90
      TT=(FLOAT(IFIX(T/100.))*60.)+(T-(FLOAT(IFIX(T/100.)))*100.)
      TT=TT-SRISE
      FDT=TT/DL
      IF(FDT.LT.0.0) FDT=0.0
      KK=0
   60 KK=KK+1
      IF(FDT.GE.FD(KK).AND.FDT.LT.FD(KK+1)) GO TO 70
      IF(KK.LT.5) GO TO 60
      IF(FDT.GE.FD(6)) GO TO 90
   70 CONTINUE
      X=FDT-FD(KK)
      FG=((A(KK)*X+B(KK))*X +C(KK))*X+Y(KK)
      AMIX(J)=HMIN+FG*DELH
      NMIX=NMIX+1
      IF (T.GE.FLOAT(JJSTP)) GO TO 120
      GO TO 100
   80 AMIX(J)=Z1
      NMIX=NMIX+1
      GO TO 100
   90 AMIX(J)=Z2
      NMIX=NMIX+1
  100 CONTINUE
      T=T+100.
  110 CONTINUE
  120 CONTINUE
      IF(DEND.GT.TTMAX) DEND=TTMAX
      RETURN
      END
      SUBROUTINE RATES (C,N)
      SAVE
C
C  SETUP RATE CONSTANTS FOR THE CURRENT CHEMICAL MECHANISM
C
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),CR(61),REACT(61),NI,PLSP(5),KOZ(5)
      DIMENSION C(1)
      DIMENSION SIG(61)
      INTEGER SPECIS,REACT,HCSPEC
      FCT=(1./298.)-(1./TEMP)
      DO 10 I=1,51
   10 SIG(I)=0.
      IF (N.LE.0) GO TO 40
      DO 30 I=1,N
      DO 20 J=1,NS
      IF (SPECIS(J).EQ.REACT(I)) SIG(J)=C(I)
      IF (SPECIS(J).EQ.REACT(I)) GO TO 30
   20 CONTINUE
      SIG(NS+1)=SIG(NS+1)+C(I)
   30 CONTINUE
   40 N=NS
      M=N-1
      C(N)=0.
      DO 50 I=1,M
      C(N)=C(N)+SIG(I)
   50 C(I)=SIG(I)
      BK=0.
      C(N)=C(N)+SIG(NS+1)
      IF (SIG(N).NE.0.) BK=SIG(N)-C(N)
      IF (SIG(N).NE.0.) C(N)=SIG(N)
      NP=0
      DO 70 I=1,NR
      IF (KR(I,1).EQ.0) GO TO 70
      ITYPE(I)=2
      IF (KR(I,1).EQ.99.AND.KR(I,3).EQ.99) KR(I,1)=N
      IF (KR(I,1).EQ.N.AND.KR(I,3).EQ.99) KR(I,3)=0
      IF (KR(I,2).EQ.0.AND.KR(I,3).EQ.99) KR(I,2)=N
      IF (KR(I,2).EQ.N.AND.KR(I,3).EQ.99) KR(I,3)=0
      IF (KR(I,2).EQ.0) ITYPE(I)=1
      IF (KR(I,3).NE.0) ITYPE(I)=3
      IF (KR(I,3).EQ.99) KR(I,3)=N
      DO 60 J=4,12
      IF (KR(I,J).LT.0) KR(I,J)=-KR(I,J)
   60 CONTINUE
      IF (ABS(S(I)).EQ.0.) R(I)=A(I)
      IF (ABS(S(I)).NE.0.) R(I)=A(I)*EXP(S(I)*FCT)
   70 CONTINUE                                                         I     51
      RETURN                                                           I     52
      END                                                              I     53-
      SUBROUTINE SPARS (IA,JA,N)
      SAVE
C
C  SETUP INTERNAL GEAR POINTER VECTOR
C
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IB(60)
     1,JB(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      DIMENSION IA(N), JA(N)
      DO 10 I=1,N
   10 IA(I)=1
      KT=0
      IA(N+1)=1
      JA(1)=0
      DO 140 IR=1,NR
      IF (KR(IR,1).EQ.0.OR.KR(IR,1).EQ.99) GO TO 140
      MT=ITYPE(IR)
      DO 130 K=1,MT
      I=KR(IR,K)
      DO 60 L=1,MT
      J=KR(IR,L)
      K1=IA(J)
      K2=IA(J+1)-1
      IF (K1.GT.K2) GO TO 30
      DO 20 M=K1,K2
      IF (I.EQ.JA(M)) GO TO 60
   20 CONTINUE
   30 DO 40 M=J,N
   40 IA(M+1)=IA(M+1)+1
      KT=KT+1
      KD=KT-K2
      K2=K2+1
      DO 50 M=1,KD
         KTM = KT-M
         JA(KTM+2) = JA(KTM+1)
   50 CONTINUE
      JA(K2)=I
   60 CONTINUE
      K1=IA(I)
      DO 120 L=4,12
      K2=IA(I+1)-1
      J=KR(IR,L)
      IF (J) 120,130,70
   70 IF (K1.GT.K2) GO TO 90
      DO 80 M=K1,K2
      IF (J.EQ.JA(M)) GO TO 120
   80 CONTINUE
   90 DO 100 M=I,N
```

```
  100 IA(M+1)=IA(M+1)+1
      KT=KT+1
      KD=KT-K2
      K2=K2+1
      DO 110 M=1,KD
         KTM = KT-M
         JA(KTM+2) = JA(KTM+1)
  110 CONTINUE
      JA(K2)=J
  120 CONTINUE
  130 CONTINUE
  140 CONTINUE
      DO 160 I=1,N
      K1=IA(I)+1
      K2=IA(I+1)-1
      IF (K1.GT.K2) GO TO 160
      MT=K2-K1+1
      DO 150 K=1,MT
      DO 150 M=K1,K2
      IF (JA(M).GT.JA(M-1)) GO TO 150
      J=JA(M-1)
      JA(M-1)=JA(M)
      JA(M)=J
  150 CONTINUE
  160 CONTINUE
      M=N
      DO 190 I=1,M
      IF(IA(I+1).GT.IA(I)) GO TO 190
      NM=I+1
      NN=N+1
      KMIN=IA(NM)
      KMAX=IA(NN)
      DO 170 J=KMIN,KMAX
      KM=KMAX+KMIN-J
  170 JA(KM)=JA(KM-1)
      KNOW=IA(I)
      JA(KNOW)=I
      DO 180 LL=NM,NN
  180 IA(LL)=IA(LL)+1
  190 CONTINUE
      RETURN
      END
      SUBROUTINE CONVT (NUM,L,N)
      SAVE
C
C  SUBROUTINE CONVT CONVERTS INTEGERS TO ALPHANUMERICS
C     FOR PRINTING
C
C  ASSUMES VALUE OF INTEGER IS POSITIVE
C
      DIMENSION L(5), JDIGIT(10)
C
      DATA JDIGIT/'0','1','2','3','4','5','6','7','8','9'/
      DATA JBLANK/' '/
C
      NI=NUM
      DO 10 I=1,N
      L(I)=JBLANK
   10 CONTINUE
C
      DO 20 K=1,N
      I=N-K+1
      NEXT=NI/10
      NDX=(NI-NEXT*10)+1
      L(I)=JDIGIT(NDX)
      IF (NEXT.LE.0) GO TO 30
      NI=NEXT
   20 CONTINUE
C
   30 RETURN
      END
      SUBROUTINE VALU (VAL,N,NF,NL)
      SAVE
C
C  CONVERT NUMERIC VALUES TO ALPHAS (ONLY USED TO WRITE MECHANISM)
C
      COMMON /STORE/ AST(60)
      COMMON /GEAR1/ T,H,HMIN,HMAX,EPS1,UROUND,NC,MF1,KFLAG1,JSTART
      DIMENSION AC(5), AD(5), BC(5)
      INTEGER AC,AD,BC,AST
      DATA IBLK/' '/,IPER/'.'/,IZRO/'0'/
      VAL1=ABS(VAL)
      AVAL=ALOG10(VAL1)
      IF (AVAL.EQ.0.) RETURN
      DO 10 I=1,5
      AC(I)=IBLK
      AD(I)=IZRO
   10 BC(I)=IZRO
      IAD=IFIX(AVAL+UROUND)
      IF (IAD) 60,20,20
   20 IREM=3-IAD
      IAD=IAD+1
      JREM=IFIX(VAL1+UROUND)
      REM=VAL1-FLOAT(JREM)
      IF (REM.GT.UROUND.AND.IREM.GT.0) GO TO 30
      CALL CONVT (JREM,AC,5)
      GO TO 100
   30 CALL CONVT (JREM,AD,IAD)
      JREM=IFIX(REM*(10.**IREM)+0.1)
      CALL CONVT (JREM,BC,IREM)
      DO 40 J=1,IAD
   40 AC(J)=AD(J)
      AC(IAD+1)=IPER
      DO 50 K=1,IREM
          KIAD1 = K+IAD+1
      AC(KIAD1)=BC(K)
   50 CONTINUE
      GO TO 80
   60 IAD=IABS(IFIX(AVAL-0.1))-1
      IVAL=IFIX(VAL1*10000.+0.1)
      CALL CONVT (IVAL,AC,5)
      AC(1)=IPER
      IF (ABS(VAL1+UROUND-0.1).LT.0.00001) GO TO 80
      DO 70 I=1,IAD
   70 AC(I+1)=IZRO
   80 IF (AC(5).NE.IZRO) GO TO 100
      DO 90 K=1,4
      L=6-K
   90 AC(L)=AC(L-1)
      IF (AC(1).EQ.IPER) IVV=IZRO
      IF (AC(1).NE.IPER) IVV=IBLK
      AC(1)=IVV
      GO TO 80
  100 DO 110 I=NF,NL
      K=I-NF+1
  110 AST(I)=AC(K)
      RETURN
      END
      SUBROUTINE EMISS (N,H,COEFF)
      SAVE
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C TO HANDLE MORE THAN 16 HISTOGRAM INTERVALS (ASSUMING N IS #INTERVALS)
C THE DIMENSIONED VARIABLES BELOW MUST HAVE THE FOLLOWING DIMENSIONS
C H = N; Y,DYDX = N+1; CF,COEFF = 5*N
C SEE ALSO DIMENSION STATEMENT IN STAIR
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
        DIMENSION H(25), CF(125), Y(26), DYDX(26), COEFF(125)
        DOUBLE PRECISION CF, Y, DYDX, HRATIO, EXPONE, FORRAT, FORCDY
        DOUBLE PRECISION SIGNDY, SMALLV, RATLIM
        REAL DELX, H, COEFF
        INTEGER STARTY, STOPY, INDEX, N, MODE, MODE1I
        INTEGER NPLUS1, FIVEN, J, K
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C   N IS THE NUMBER OF HISTOGRAM INTERVALS
C   N + 1  IS THE NUMBER OF EDGES IN THE HISTOGRAM PLOT
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
        RATLIM= 5.001
        MODE=1
        DELX=1.0
        FORRAT= 2.5
        FORCDY = 5.25
        NPLUS1 = N + 1
        FIVEN = 5 * N
        IF ( FIVEN .LE. 1) GO TO 20
        DO 10 J = 1 , FIVEN
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C   SET ALL COEFFICIENTS TO ZERO IN THE CF ARRAY.
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
          CF(J) = 0.0
   10   CONTINUE
   20   CONTINUE
        IF ( NPLUS1 .LE. 1) GO TO 40
        DO 30 J = 1 , NPLUS1
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  ZERO OUT ALL THE Y'S AND DYDX'S,  IN THE PROCESS
C    FORCING THE EDGE OF THE LAST DELX INTERVAL TO ZERO
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
          Y(J) = 0.0
          DYDX(J) = 0.0
C
```

```
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  LOOK FOR THE FIRST NON-ZERO H
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
   30  CONTINUE
   40  CONTINUE
       INDEX = 1
C      WHILE ((H(INDEX) .LE. 0.0) .AND. (INDEX .LT. N))
   50  IF ((H(INDEX) .GT. 0.0) .OR. (INDEX .GE. N)) GO TO 60
         INDEX = INDEX + 1
       GO TO 50
   60  CONTINUE
       STARTY = INDEX
       INDEX = INDEX + 1
       MODE1I = 1
C      IF (MODE .NE. 0)
       IF (MODE .EQ. 0) GO TO 70
C       THEN
         MODE1I = STARTY
         Y(STARTY) = H(STARTY)
         DYDX(STARTY) =  0.0
   70  CONTINUE
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  STAIR DETERMINES THE COEFFICIENTS OF THE FOURTH DEGREE
C  POLYNOMIAL F(X)= (((AX+B)X+C)X+D)X+E SUCH THAT F(X) IS
C  TWICE DIFFERENTIABLE ACROSS HISTOGRAM INTERVAL EDGES,
C  MATCHES IN VALUE AT THE INTERVAL EDGES, AND SUCH THAT
C  THE INTEGRAL OF F(X)DX = THE HISTOGRAM INTERVAL HEIGHT
C  TIMES THE INTERVAL WIDTH (DELX). THE CROSSOVER POINTS
C  (THE VALUES OF F(X) AT THE EDGES BETWEEN TWO HISTOGRAM
C  INTERVALS) ARE THE KNOWNS. STAIR SOLVES FOR THE VALUES
C  OF F'(X) AT THE EDGES GIVEN THE BOUNDARY CONDITIONS OF
C  F'(X) AT THE TWO ENDS OF EACH STAIR EVALUATION. HISFIT
C  STEPS THROUGH THE HISTOGRAM IN ONE PASS, BREAKING THE
C  HISTOGRAM INTO STAIR EVALUATIONS OF 1 OR MORE INTERVALS.
C  NOTE THAT THE CURVE DEFINED BY THE COEFFICIENTS IS ONLY
C  ONCE DIFFERENTIABLE AT EDGES BETWEEN STAIR EVALUATIONS.
C  STAIR EVALUATIONS MAY NOT CROSS POINTS WHERE F'(X) IS
C  KNOWN OR FIXED E.G. ZERO POINTS (DY/DX = 0), OR FORCE
C  FITS (WHERE UNRESTRAINED STAIR FITS RESULT IN FUNCTIONS
C  WITH NEGATIVE VALUES WHICH CORRESPONDS TO NO PHYSICAL
C  REALITY). AT THESE EDGES, THE CURRENT STAIR EVALUATION
C  IS TERMINATED AND A NEW STAIR EVALUATION STARTED. IF
C  THE RATIO OF ADJACENT HISTOGRAM VALUES IS LESS THAN 5/1
C  A CROSSOVER POINT IS PICKED ACCORDING TO THE HEURISTIC:
C         Y =  (RATIO ** 0.50)*(SMALLER VALUE)
C  WHERE 5.001 .GT. RATIO .GT. 1. FOR THE SPECIAL CASE OF A VALUE
C  WHICH HAS A ZERO VALUE TO ONE SIDE, THE OTHER EDGE
C  CROSSOVER IS: Y =  (RATIO ** 0.57)*(SMALLER VALUE)
```

```
C  WHERE 5.001 .GT. RATIO .GT. 1. IF THE RATIO OF ADJACENT
C  HISTOGRAM VALUES EXCEEDS 5 A FIT IS FORCED BY APPROX-
C  IMATING A RATIO-OF-5 FIT THROUGH THE SMALLER VALUE.
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C         WHILE (INDEX .LE. N)
   80    IF (INDEX .GT. N) GO TO 240
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C   STEP THROUGH HISTOGRAM, BREAK INTO STAIR EVALUATIONS
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C              IF (H(INDEX) .LE. 0.0)
               IF (H(INDEX) .GT. 0.0) GO TO 110
C               THEN
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  CASE OF NEGATIVE OR ZERO HEIGHT
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
                 STOPY = INDEX
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  FIT A SPLINE TO THIS POINT
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
                 CALL STAIR(STARTY,STOPY,DELX,H,Y,DYDX,CF)
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  LOOK FOR NEXT NON ZERO H
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C                WHILE ((H(INDEX) .LE. 0.0) .AND. (INDEX .LT. N))
   90            IF ((H(INDEX) .GT. 0.0) .OR. (INDEX .GE. N))
     1              GO TO 100
                  INDEX = INDEX + 1
                 GO TO 90
  100            CONTINUE
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  START NEW STAIR EVALUATION
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
                 STARTY = INDEX
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C   GOTO BOTTOM OF LOOP
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
                 GO TO 230
```

```
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  IT IS GUARANTEED THAT H(INDEX-1) AND H(INDEX) ARE .GT. 0.0
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
  110      CONTINUE
C          ELSE
            HRATIO = H(INDEX) / H(INDEX-1)
            EXPONE= 0.50
C           IF (HRATIO .GE. 1.0)
            IF (HRATIO .LT. 1.0) GO TO 160
C            THEN
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  BEGIN UP STEP
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
              SIGNDY = 1.0
              SMALLV= H(INDEX-1)
C             IF ((INDEX .EQ. (MODE1I + 1)) .AND. (MODE .EQ. 0))
              IF ((INDEX .NE. (MODE1I + 1)) .OR. (MODE .NE. 0))
     1           GO TO 120
C              THEN
                EXPONE = 0.57
                GO TO 150
  120          CONTINUE
C              ELSE
C               IF .NOT.((INDEX .EQ. 2) .AND. (MODE .NE. 0))
                IF ((INDEX .EQ. 2) .OR. (MODE .NE. 0)) GO TO 140
C                THEN
C                 IF (H(INDEX-2) .EQ. 0.0)
                  IF (H(INDEX-2) .NE. 0.0) GO TO 130
C                  THEN
                    EXPONE = 0.57
  130             CONTINUE
  140           CONTINUE
  150         CONTINUE
              GO TO 200
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  END UP STEP
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
  160         CONTINUE
C             ELSE
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  BEGIN DOWN STEP
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
```

```
              HRATIO = 1.0 / HRATIO
              SIGNDY = -1.0
              SMALLV= H(INDEX)
C             IF (INDEX .EQ. N)
              IF (INDEX .NE. N) GO TO 170
C              THEN
                EXPONE = 0.57
                GO TO 190
  170          CONTINUE
C              ELSE
C               IF (H(INDEX+1) .EQ. 0.0)
                IF (H(INDEX+1) .NE. 0.0) GO TO 180
C                THEN
                  EXPONE = 0.57
  180           CONTINUE
  190         CONTINUE
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  END DOWN STEP
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
  200       CONTINUE
C           IF (HRATIO .LT. RATLIM)
            IF (HRATIO .GE. RATLIM) GO TO 210
C            THEN
              Y(INDEX) = (HRATIO ** EXPONE) * SMALLV
              GO TO 220
C            ELSE
  210        CONTINUE
              Y(INDEX) = FORRAT * SMALLV
              DYDX(INDEX) = SIGNDY * FORCDY * (Y(INDEX) / DELX)
              STOPY = INDEX
              CALL STAIR(STARTY,STOPY,DELX,H,Y,DYDX,CF)
              STARTY = INDEX
  220       CONTINUE
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C BOTTOM OF LOOP, EXAMINE NEXT INTERVAL
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
  230     CONTINUE
          INDEX = INDEX + 1
        GO TO 80
  240   CONTINUE
C       ENDWHILE (INDEX .LE. N) LOOP
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  FOR THE FINAL SPLINE PORTION, CALL STAIR.
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
```

```
C        IF (STARTY .NE. (N+1)) .AND. (H(N) .NE. 0.0)
         IF ((STARTY .EQ. (N+1)) .OR. (H(N) .EQ. 0.0))
     1      GO TO 250
C         THEN
           STOPY = N + 1
           CALL STAIR(STARTY,STOPY,DELX,H,Y,DYDX,CF)
  250    CONTINUE
         IF (1 .GT. FIVEN) GO TO 270
         DO 260 K = 1 , FIVEN
           COEFF(K) = CF(K)
  260    CONTINUE
  270    CONTINUE
       RETURN
       END
```

```
      SUBROUTINE STAIR(KSTRT,KSTOP,DELX,H,Y,DYDX,CF)
      SAVE
C
C -----------------------------------------------------------------------
C TO HANDLE MORE THAN 16 HISTOGRAM INTERVALS (ASSUMING N IS #INTERVALS)
C THE DIMENSIONED VARIABLES BELOW MUST HAVE THE FOLLOWING DIMENSIONS
C H = N; Y,DYDX,A,B,C = N+1; CF = 5*N
C SEE ALSO DIMENSION STATEMENT IN HISFIT
C -----------------------------------------------------------------------
C
        DIMENSION A(26), B(26), C(26), CF(125), Y(26), DYDX(26), H(25)
        DOUBLE PRECISION A, B, C, R, S1, S2, T1, T2, U1, U2, V1, V2, Z
        DOUBLE PRECISION Y, DYDX, CF
        REAL DELX, H
        INTEGER K,KSTRT,KSTOP,KSTOP1, J1, J2
        INTEGER KSTOP2, KDOWN, KSTRT1
        KSTOP1 = KSTOP - 1
C
C -----------------------------------------------------------------------
C  COMPUTE COEFFICIENTS FOR
C   FK(X)=AK(X-XK)4 + BK(X-XK)3 + CK(X-XK)2 + DK(X-XK) +EK
C
C  STORE IN ARRAY CF AS:
C    CF 5*K =A;       CF 5*K-1 =B;       CF 5*K-2 =C;
C    CF 5*K-3 =D=YKP                     CF 5*K-4 =E=YK
C    CF 5*K-3 =D=YKP                     CF 5*K-4 =E=YK
C     FOR K=KSTRT TO KSTOP  (WHERE KSTOP-KSTRT IS THE NUMBER
C                                OF INTERVALS. )
C   KSTRT  IS THE LEFT EDGE OF THE STARTING HISTOGRAM BAR
C   KSTOP  IS THE RIGHT EDGE OF THE STOPPING HISTOGRAM BAR
C -----------------------------------------------------------------------
C
        IF (KSTRT .GT. KSTOP1) GO TO 20
        DO 10 K = KSTRT , KSTOP1
          CF(5*K-4)=Y(K)
   10   CONTINUE
   20   CONTINUE
C
C -----------------------------------------------------------------------
C SET UP AND SOLVE MATRIX
C -----------------------------------------------------------------------
C
        B(KSTRT)=0.0
        C(KSTRT)=0.0
        IF ( KSTRT .GT. KSTOP1) GO TO 70
        DO 60 K = KSTRT , KSTOP1
          J2=K+1
          U2=1.0/(DELX)
          A(K)=U2
          V2=U2*U2
          S2=20.0*V2*H(K)
          T2=8.0*V2*(Y(K)+Y(J2))
C         IF (K.GT.KSTRT)
          IF (K.LE.KSTRT) GO TO 50
C          THEN
            Z=1.0/(3.0*(U1+U2)+U1*B(J1))
            B(K)= -U2*Z
            R=S2-S1-T2+T1+4.0*(V1-V2)*Y(K)
C           IF (K.EQ.KSTRT+1)
            IF (K.NE.KSTRT+1) GO TO 30
C            THEN
              R=R+U1*DYDX(KSTRT)
   30       CONTINUE
C           IF (K.EQ.KSTOP1)
            IF (K.NE.KSTOP1) GO TO 40
C            THEN
              R=R+U1*DYDX(KSTOP)
   40       CONTINUE
            C(K)=Z*(R+U1*C(J1))
   50     CONTINUE
          J1=K
          U1=U2
          V1=V2
          S1=S2
          T1=T2
   60   CONTINUE
   70   CONTINUE
C
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C  BACK SUBSTITUTION
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C       IF (KSTOP-KSTRT.GT.1)
        IF (KSTOP-KSTRT.LE.1) GO TO 100
C        THEN
          DYDX(KSTOP-1)=C(KSTOP-1)
          KSTOPP=KSTOP-1
          CF(5*KSTOPP-3)=DYDX(KSTOPP)
C         IF (KSTOP-KSTRT.GT.2)
          IF (KSTOP-KSTRT.LE.2) GO TO 90
C          THEN
            KSTOP2 = KSTOP - 2
            KSTRT1 = KSTRT + 1
            DO 80 K = KSTRT1 , KSTOP2
              KDOWN = KSTRT1 + KSTOP2 - K
              DYDX(KDOWN) = C(KDOWN) - B(KDOWN) * DYDX(KDOWN+1)
              CF(5*KDOWN-3) = DYDX(KDOWN)
   80       CONTINUE
   90      CONTINUE
  100    CONTINUE
         CF(5*KSTRT-3)=DYDX(KSTRT)
C
C -----------------------------------------------------------------
C COMPUTE 2ND, 3RD, 4TH ORDER COEFFICIENTS
C -----------------------------------------------------------------
C
         IF ( KSTRT .GT. KSTOP1) GO TO 120
         DO 110 K = KSTRT , KSTOP1
           J2=K+1
           Z=A(K)
           CF(5*K-2)= 1.5*Z*(-3.0*DYDX(K) + DYDX(J2) + Z*(-8.0*
     1     Y(J2) - 12.0*Y(K) + 20.0*H(K)))
           CF(5*K-1)= -4.0*Z*Z*(-1.5*DYDX(K) + DYDX(J2) + Z*(-7.0*
     1     Y(J2) - 8.0*Y(K) +15.0*H(K)))
           CF(5*K) = 5.0*Z*Z*Z*(0.5*(DYDX(J2)-DYDX(K)) - 3.0*Z*
     1     (Y(J2) + Y(K) - 2.0*H(K)))
  110    CONTINUE
  120    CONTINUE
       RETURN
       END
```

```
      SUBROUTINE PHOT
      SAVE
C
C    ****    RATE CONSTANT CALCULATIONS FOR FIRST ORDER PHOTOCHEMICAL
C            REACTIONS
C            REF: SCHERE AND DEMERJIAN (1977)
C
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),K(24),JSTRT,JSTOP,SPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      INTEGER SPEC
      DIMENSION D(2), C(72), W(72), V(5), TMS(25)
      DATA D/0.,0./
C
      IEND=24
      NTM=IEND
      TIME=JSTRT
      DO 10 II=1,IEND
C
C    ****    CALL SUBROUTINE TO COMPUTE ZENITH ANGLES FROM TIME OF DAY
C
      XC=0.0
      TSTD=TIME
      IF (TSTD.LT.0.) TSTD=TSTD+2400.
      CALL SOLAR (SLA,SLO,TZ,IY,IM,ID,TSTD,XC,5)
      ITIME(II)=TIME
      XZ(II)=90.-XC
      TIME=CLOCK(TIME,IINC)
C
      IF (TIME.GT.2400.) TIME=TIME-2400.
   10 CONTINUE
C
      DO 30 L=1,IP
      ISTRT=(MNLM-LAM1)/INC+1
      ISTOP=(MXLM-LAM1)/INC+1
C
C    ****    CALL FIRST SUBROUTINE FOR SPLINE INTERPOLATION OF RATE CONS
C
      IF (L.EQ.1) CALL SPLNA (MAXZ,Z,RTCON,2,D,C,W)
      IF (L.GT.1) CALL SPLNA (10,Z,PP(1,L),2,D,C,W)
      DO 20 II=1,IEND
      V(1)=XZ(II)
      P(II,L)=0.
      IF (V(1).GT.90.0) GO TO 20
C
C    ****    CALL SECOND SUBROUTINE IN SPLINE INTERPOLATION SCHEME
C
```

```
      IF (L.EQ.1) CALL SPLNB (MAXZ,Z,RTCON,C,V)
      IF (L.GT.1) CALL SPLNB (10,Z,PP(1,L),C,V)
      P(II,L)=AMAX1(0.0,V(2))
   20 CONTINUE
C
   30 CONTINUE
C     DETERMINE TIME OF SOLAR NOON
C
      IZMN=IEND
      XZEM=9999.
      DO 40 I=1,IEND
      IF (XZ(I).GE.XZEM) GO TO 40
      XZEM=XZ(I)
      IZMN=I
   40 CONTINUE
      I=IZMN
   50 TIME=ITIME(I-1)
      XA=XZ(I-1)
      DO 60 I=1,60
      SPECIE=TIME
      TIME=CLOCK(TIME,1)
      TSTD=TIME
      IF (TSTD.LT.0.) TSTD=TSTD+2400.
      XB=XA
      CALL SOLAR (SLA,SLO,TZ,IY,IM,ID,TSTD,XC,5)
      XA=90.-XC
      IF (XA.GT.XB) GO TO 70
   60 CONTINUE
   70 DO 80 I=1,IEND
      TMS(I)=60.*FLOAT(I-1)
   80 CONTINUE
      DO 100 L=1,IP
      DO 90 I=2,IEND
      NMM=IEND+2-I
      IF (P(NMM-1,L).GT.0.) GO TO 100
   90 CONTINUE
  100 CALL SPLNA (NMM,TMS,P(1,L),2,D,CF(1,L),W)
      RETURN
      END
      SUBROUTINE SOLAR (SLA,SLO,TZ,IY,IM,ID,TIME,D,NV)
      SAVE
C***
C***     SLA...  LATITUDE (DEG)  SOUTH = MINUS
C***     SLO...  LONGITUDE (DEG)  EAST = MINUS
C***     TZ...  TIME ZONE
C***            ALSO INCLUDES FRACTION IF LOCAL TIME IS NOT
C***             STANDARD MERIDIAN TIME.  E.G. POONA, INDIA 5.5
C***     IY..  YEAR
C***     IM..  MONTH
C***     ID..  DAY
C***     TIME.. LOCAL STANDARD TIME IN HOURS AND MINUTES.
C***            1 30 PM = 1330  ** STANDARD TIME **
C***     D..  RETURNED VALUE
C***     NV..  VALUE TO BE RETURNED, SELECTED AS FOLLOWS....
C***           1...  DECLINATION (DEG.)
C***           2...  EQUATION OF TIME ADJUSTMENT (HRS.)
C***           3...  TRUE SOLAR TIME (HRS.)
C***           4...  HOUR ANGLE (DEG.)
C***           5...  SOLAR ELEVATION (DEG.)
C***           6...  OPTICAL AIRMASS
C***      0 ( NV ( 7.  OTHERWISE, D = 9999.
C***
      DIMENSION MD(11)
      DATA MD/31,29,31,30,31,30,2*31,30,31,30/
      DATA A,B,C,SIGA/0.15,3.885,1.253,279.9348/
      RAD=572957.75913E-4
      SDEC=39784.988432E-5
      RE=1.
      IF (SLO.LT.0.) RE=-1.
      KZ=TZ
      TC=(TZ-FLOAT(KZ))*RE
      TZZ=FLOAT(KZ)*RE
      SLB=SLA/RAD
      K=ID
      TIMH=TIME/100.
      I=TIMH
      TIMLOC=(TIMH-FLOAT(I))/0.6+FLOAT(I)+TC
      IMC=IM-1
      IF (IMC.LT.1) GO TO 20
      DO 10 I=1,IMC
   10 K=K+MD(I)
   20 LEAP=1
      NL=MOD(IY,4)
      IF (NL.LT.1) LEAP=2
      SMER=TZZ*15.
      TK=((SMER-SLO)*4.)/60.
      KR=1
      IF (K.GE.61.AND.LEAP.LT.2) KR=2
      DAD=(TIMLOC+TZZ)/24.
      DAD=DAD+FLOAT(K-KR)
      DF=DAD*360./365.242
      DE=DF/RAD
      DESIN=SIN(DE)
      DECOS=COS(DE)
      DESIN2=SIN(DE*2.)
      DECOS2=COS(DE*2.)
      SIG=SIGA+DF+1.914827*DESIN-0.079525*DECOS+0.019938*DESIN2-0.00162*
     1DECOS2
      SIG=SIG/RAD
      DECSIN=SDEC*SIN(SIG)
      EFFDEC=ASIN(DECSIN)
      IF (NV.NE.1) GO TO 30
      D=EFFDEC*RAD
      RETURN
   30 EQT=0.12357*DESIN-0.004289*DECOS+0.153809*DESIN2+0.060783*DECOS2
      IF (NV.NE.2) GO TO 40
      D=EQT
      RETURN
   40 TST=TK+TIMLOC-EQT
      IF (NV.NE.3) GO TO 50
      D=TST
      IF (D.LT.O.) D=D+24.
      IF (D.GE.24.) D=D-24.
      RETURN
   50 HRANGL=ABS(TST-12.)*15.
      IF (NV.NE.4) GO TO 60
      D=HRANGL
      RETURN
   60 HRANGL=HRANGL/RAD
      SOLSIN=DECSIN*SIN(SLB)+COS(EFFDEC)*COS(SLB)*COS(HRANGL)
      SOLEL=ASIN(SOLSIN)*RAD
      IF (NV.NE.5) GO TO 70
      D=SOLEL
      RETURN
   70 IF (NV.NE.6) GO TO 80
      IF (SOLEL.LE.O.) GO TO 80
      TK=SOLEL+B
      E=1./TK**C
      D=1./(A*E+SOLSIN)
      RETURN
   80 D=9999.
      RETURN
      END
      SUBROUTINE SPLNA (N,X,Y,J,D,C,W)                                  Q
      SAVE                                                              Q
      DIMENSION X(24), Y(24), D(2), C(72), W(72)                        Q
C     ------------------------------------------------------------------Q
C           OVER THE INTERVAL X(I) TO X(I+1), THE INTERPOLATING         Q
C           POLYNOMIAL                                                  Q
C                Y=Y(I)+A(I)*Z+B(I)*Z**2+E(I)*Z**3                      Q
C           WHERE Z=(X-X(I))/(X(I+1)-X(I))                              Q
C           IS USED. THE COEFFICIENTS A(I),B(I) AND E(I) ARE COMPUTED Q
C           BY SPLNA AND STORED IN LOCATIONS C(3*I-2),C(3*I-1) AND      Q
C           C(3*I) RESPECTIVELY.                                        Q
C           WHILE WORKING IN THE ITH INTERVAL,THE VARIABLE Q WILL       Q
C           REPRESENT Q=X(I+1) - X(I), AND Y(I) WILL REPRESENT          Q
C           Y(I+1)-Y(I)                                                 Q
C     ------------------------------------------------------------------Q
C                                                                       Q
      Q=X(2)-X(1)                                                       Q
      YI=Y(2)-Y(1)                                                      Q
      IF (J.EQ.2) GO TO 10                                              Q
C     ------------------------------------------------------------------Q
C           IF THE FIRST DERIVATIVE AT THE END POINTS IS GIVEN,         Q
C           A(1) IS KNOWN, AND THE SECOND EQUATION BECOMES              Q
C           MERELY B(1)+E(1)=YI - Q*D(1).                               Q
C     ------------------------------------------------------------------Q
      C(1)=Q*D(1)                                                       Q
      C(2)=1.0                                                          Q
      W(2)=YI-C(1)                                                      Q
      GO TO 20                                                          Q
C     ------------------------------------------------------------------Q
C           IF THE SECOND DERIVATIVE AT THE END POINTS IS GIVEN         Q
C           B(1) IS KNOWN, THE SECOND EQUATION BECOMES                  Q
C           A(1)+E(1)=YI-0.5*Q*Q*D(1). DURING THE SOLUTION OF           Q
C           THE 3N-4 EQUATIONS,A1 WILL BE KEPT IN CELL C(2)             Q
C           INSTEAD OF C(1) TO RETAIN THE TRIDIAGONAL FORM OF THE       Q
C           COEFFICIENT MATRIX.                                         Q
C     ------------------------------------------------------------------Q
   10 C(2)=0.0                                                          Q
      W(2)=0.5*Q*Q*D(1)                                                 Q
   20 M=N-2                                                             Q
      IF (M.LE.0) GO TO 40                                              Q
C     ------------------------------------------------------------------Q
C           UPPER TRIANGULARIZATION OF THE TRIDIAGONAL SYSTEM OF        Q
C           EQUATIONS FOR THE COEFFICIENT MATRIX FOLLOWS--              Q
C     ------------------------------------------------------------------Q
      DO 30 I=1,M                                                       Q
      AI=Q                                                              Q
      Q=X(I+2)-X(I+1)                                                   Q
      H=AI/Q                                                            Q
      C(3*I)=-H/(2.0-C(3*I-1))                                          Q
      W(3*I)=(-YI-W(3*I-1))/(2.0-C(3*I-1))                              Q
      C(3*I+1)=-H*H/(H-C(3*I))
      W(3*I+1)=(YI-W(3*I))/(H-C(3*I))
      YI=Y(I+2)-Y(I+1)
      C(3*I+2)=1.0/(1.0-C(3*I+1))
   30 W(3*I+2)=(YI-W(3*I+1))/(1.0-C(3*I+1))
C     ------------------------------------------------------------------
C                E(N-1) IS DETERMINED DIRECTLY FROM THE LAST EQUATION
C                OBTAINED ABOVE, AND THE FIRST OR SECOND DERIVATIVE
C                VALUE GIVEN AT THE END POINT.
C     ------------------------------------------------------------------
   40 IF (J.EQ.1) GO TO 50
      C(3*N-3)=(Q*Q*D(2)/2.0-W(3*N-4))/(3.0-C(3*N-4))
      GO TO 60
   50 C(3*N-3)=(Q*D(2)-YI-W(3*N-4))/(2.0-C(3*N-4))
   60 M=3*N-6
      IF (M.LE.0) GO TO 80
C     ------------------------------------------------------------------
C                BACK SOLUTION FOR ALL COEFFICENTS EXCEPT
C                A(1) AND B(1) FOLLOWS--
C     ------------------------------------------------------------------
      DO 70 II=1,M
      I=M-II+3
   70 C(I)=W(I)-C(I)*C(I+1)
   80 IF (J.EQ.1) GO TO 90
C     ------------------------------------------------------------------
C                IF THE SECOND DERIVATIVE IS GIVEN AT THE END POINTS,
C                A(1) CAN NOW BE COMPUTED FROM THE KNOWN VALUES OF
C                B(1) AND E(1). THEN A(1) AND B(1) ARE PUT INTO THEIR
C                PROPER PLACES IN THE C ARRAY.
C     ------------------------------------------------------------------
      C(1)=Y(2)-Y(1)-W(2)-C(3)
      C(2)=W(2)
      RETURN
   90 C(2)=W(2)-C(3)
      RETURN
      END
      SUBROUTINE SPLNB (N,X,Y,C,V)
      SAVE
      DIMENSION X(24), Y(24), C(72), V(5)
      V(5)=2.0
      LIM=N-1
C                   -----------------------------------------------------
C                   DETERMINE IN WHICH INTERVAL THE INDEPENDENT
C                   VARIABLE,V(1),LIES.
C                   -----------------------------------------------------
      DO 10 I=2,LIM
      IF (V(1).LT.X(I)) GO TO 20
   10 CONTINUE
      I=N
      IF (V(1).GT.X(N)) V(5)=3.0
      GO TO 30
   20 IF (V(1).LT.X(1)) V(5)=1.0
C                   -----------------------------------------------------
C                   Q IS THE SIZE OF THE INTERVAL CONTAINING V(1).
C                   -----------------------------------------------------
C                   Z IS A LINEAR TRANSFORMATION OF THE INTERVAL
C                   ONTO (0,1) AND IS THE VARIABLE FOR WHICH
C                   THE COEFFICIENTS WERE COMPUTED BY SPLNA.
C                   -----------------------------------------------------
   30 Q=X(I)-X(I-1)
      Z=(V(1)-X(I-1))/Q
      V(2)=((Z*C(3*I-3)+C(3*I-4))*Z+C(3*I-5))*Z+Y(I-1)
      V(3)=((3.*Z*C(3*I-3)+2.0*C(3*I-4))*Z+C(3*I-5))/Q
      V(4)=(6.*Z*C(3*I-3)+2.0*C(3*I-4))/(Q*Q)
      RETURN
      END
      FUNCTION CLOCK (T1,IINC)
      SAVE
C
C     ****    ADD A TIME IN MINUTES TO A 2400 HOUR TIME AND RETURN A 2400
C     ****    HOUR TIME
C
      T2=IINC
      I100=INT(T1/100.)
      T3=T1-100.0*FLOAT(I100)+T2
      I100=I100+INT(T3/60.)
      CLOCK=FLOAT(I100)*100.0+T3-60.0*FLOAT(INT(T3/60.0))
      RETURN
      END
      SUBROUTINE SUNTIM
      SAVE
C
C   FIND SUNRISE AND SUNSET VALUES
C
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),KJ(24),JSTRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /MIXING/ DSTRT,DEND,AMC(5),BMC(5),CMC(5),FD(6),FG(6),
     1AMXX(26),DL,TTMAX,SSRISE,SRMIN,DELH,TDIL,NMXX,HEIGHT,SSET,SRISE
      INTEGER PSPEC
C
C   FIND SUNRISE
C
      T=0.
   10 CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.GE.0.000001) GO TO 20
      T=CLOCK(T,60)
      GO TO 10
C
C   FIND HALF HOUR VALUE
C
   20 T=T-150.
      CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.GE.0.000001) GO TO 40
C
   30 T=CLOCK(T,1)
      CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.GE.0.000001) GO TO 60
      GO TO 30
C
   40 T=T-30.
   50 T=CLOCK(T,1)
      CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.LT.0.000001) GO TO 60
      GO TO 50
C
   60 SRISE=T
C
C   FIND SUNSET
C
      T=1200.
   70 CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.LT.0.000001) GO TO 80
      T=CLOCK(T,60)
      GO TO 70
C
C   FIND HALF HOUR VALUE
C
   80 T=T-150.
```

```
      CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.LT.0.000001) GO TO 100
C
   90 T=CLOCK(T,1)
      CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.LT.0.000001) GO TO 120
      GO TO 90
C
  100 T=T-30.
  110 T=CLOCK(T,1)
      CALL SOLAR(SLA,SLO,TZ,IY,IM,ID,T,D,5)
      IF (D.GE.0.000001) GO TO 120
      GO TO 110
C
  120 SSET=T
C
      RETURN
      END
      SUBROUTINE EKCALC (X)
      SAVE
C
C     THIS SUBROUTINE PERFORMS VOC CONTROL REQUIREMENTS AS OUTLINED
C   IN THE 1981 EPA GUIDELINES ON THE USE OF CITY-SPECIFIC EKMA.  THIS
C   ROUTINE ALSO INCLUDES PROCEDURES DESCRIBED IN APPENDIX B OF THE
C   1981 GUIDELINES.
C
C    THIS ROUTINE WAS MODIFIED OCT 1986 BY HH/SAI TO:
C
C        ALLOW FOR TWO OCCURANCES OF OZONE ON A NMOC/NOX LINE
C        TO SKIP BASE YEAR CALCULATION IF LOCATION IS KNOWN
C
C    MODIFIED JUNE 1987 HH/SAI TO:
C
C        ALLOW FOR CO EMISSION CREDITS
C
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),IPRNT,TSTEP,ZENI
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),KKK(24),JSTRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      COMMON /ZENITH/ IPZ,ZDEF(10,20),IPHZ(20),IZENP
      COMMON /EMIS/ NEM,ISP,ESTRT(5),ESTOP,ESLP,EMSP(5),IEMLS(5),EOSLP(5
     1),EMO(26,5),ECI(5),EM(26),EC(125),ECO(125,5)
      COMMON /MIX/ NMIX,AMIX(26),STRM,STOPM,DC(104)
      COMMON /MIXING/ DSTRT,DEND,AMC(5),BMC(5),CMC(5),FD(6),FG(6),
     1 AMXX(26),DL,TTMAX,SSRISE,SRMIN,DELH,TDIL,NMXX,HEIGHT,SSET,SRISE
      COMMON /TEMPER/ TEMEND,NTEMP,QM(30)
      COMMON /FRPLOT/ SAVCON(80,5),SAVTIM(80),NTSV,INOW
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /TITL/ ITTL(36)
      COMMON /NEED/ HC,XN,NL,OZP(20),OZN(11,11,5),MR,LS,HCS,XNS
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,IIO3,
     1IIH20,IBZA,JPLUS
      COMMON /STORE/ AST(60)
      COMMON /HOUR/ OZM(5),NGO,TTM,TMY(5)
      COMMON /PLTVEC/ HCT(20),OT(20),NT,OHC,HCG,PLTGRD,OXN,XNG,HC1,XN1,T
     1ICZ,DIGZ,CHRZ
      COMMON /ALOFT/ IALFT,CALFT(10),LOCALF(10),ISPAL(10)
      COMMON /VVLB1/ FCTR,DIST,CHRSIZ,NNCHR,OZBL
      COMMON /BK1/ FBK(10),FBKAL(5),HCBK,XNBK,OZBK,H2OBK
      COMMON /CRED/ ICR,ISPCR,ISPNCR(3),SPCR69(3),SURFCR(3),ALOFCR(3),
     1REDCR(3),FSRFCR(3),FALFCR(3),COSFBK,COAFBK
      DIMENSION X(7),OZOUT(11),OZV(10),HCTMP(10),ANXTMP(10),
     1OZSORT(10),HCSORT(10),ANSORT(10),YP(20),TM(20),SCRSV(3),ALCRSV(3),
     2SPCRSV(3)
C
      FOZS = OZIN
      FOZA = OZAL
      FOCS = HCIN
      FOCA = HCAL
      FXNS = XNIN
      FXNA = XNAL
      CFHC = 0.0
      CFNOX = 0.0
      NPTO=0
      INFO=-1
      DOZ = 0.12
      AOZ=0.12
      HCTNOX=1.
      NTRYS = 8
      ACCU = 0.005
      RNMOC=0.60
      BNMOC = HCBK
      IF (ABS(X(1)) .NE. 0.0) DOZ = ABS(X(1))
      IF (ABS(X(2)) .NE. 0.0) HCTNOX = ABS(X(2))
      IF (X(4) .GT. 0.0) READ (IN,400) FOZS,FOZA,FOCS,FOCA,FXNS,FXNA
C
      READ (IN,400) HCMEAS,ANMEAS,ALREHC,ALRENX,HCLV
      REDNOX = 1.0 + X(3)*.01
C
C
C  DO SOME PRELIMINARY CALCULATIONS
C
      OZINSV=OZIN
      OZALSV=OZAL
      HCINSV=HCIN
      HCALSV=HCAL
      XNINSV = XNIN
      XNALSV=XNAL
      IF (ICR.LE.0) GO TO 20
      DO 10 J=1,ICR
      SCRSV(J)=SURFCR(J)
      ALCRSV(J)=ALOFCR(J)
      SPCRSV(J)=SPCR69(J)
   10 CONTINUE
C
   20 REDNX1=X(3)
C
C  ESTIMATE FUTURE NMOC IF NOT GIVEN
```

```
C
      RNMOC1=FOCA
      IF (FOCA.GE.0.) GO TO 30
      IF (FOCA.GT.(-99.)) RNMOC=ABS(FOCA)*0.01
      RNMOC1 = (HCALSV - BNMOC)*RNMOC + BNMOC
C
C  ESTIMATE FUTURE NMOC IN THE SURFACE LAYER
C
   30 RNMOC2 = (HCINSV - BNMOC)*RNMOC + BNMOC
      IF (FOCS.GE.0.) RNMOC2=FOCS
      IF (FOCS.LT.(-0.00001)) CFHC=ABS(FOCS)
      IF (FXNS.LT.(-0.00001)) CFNOX=ABS(FXNS)
      CHCNOX = HCTNOX*(1.-CFHC)/(1.-CFNOX)
      ROZ1=FOZA
      IF (FOZA.GE.0.) GO TO 50
C
C  USE THE OZONE TRANSPORT CURVES TO ESTIMATE FUTURE OZONE IF
C  FUTURE OZONE IS NEGATIVE (SEE 1981 CITY-SPECIFIC GUIDELINES P. 54)
C
      ROZ1=OZALSV
      IF (OZALSV .LE. 0.04) GO TO 50
C
C  IF INPUTED FUTURE OZONE VALUE IS LESS -100. THEN USE
C    DASH LINE OF OZONE CURVE
C
      IF (FOZA.LT.(-99.)) GO TO 40
      ROZ1 = 0.7*OZALSV + 0.012
      IF (OZALSV .GT. 0.1543) ROZ1 = 0.12
      GO TO 50
   40 ROZ1 = 0.9*OZALSV + 0.004
      IF (OZALSV .GT. 0.129) ROZ1 = 0.12
C
C  REPEAT FUTURE OZONE ESTIMATE FOR SURFACE LAYER
C
   50 ROZ2=FOZS
      IF (FOZS.GE.0.) GO TO 70
      ROZ2=OZINSV
      IF (OZINSV .LE. 0.04) GO TO 70
      IF (FOZS.LT.(-99.))  GO TO 60
      ROZ2 = 0.7*OZINSV + 0.012
      IF (OZINSV .GT. 0.1543) ROZ2 = 0.12
      GO TO 70
   60 ROZ2 = 0.9*OZINSV + 0.004
      IF (OZINSV .GT. 0.129) ROZ2 = 0.12
C
   70 HC1 = -99.0
      HC2 = 1.0
      OXN2 = HC2/HCTNOX
      IFL=0
C
```

```
      WRITE (IOUT,410) DOZ,HCTNOX,REDNX1,ROZ1,RNMOC1,FXNA
      IF (ICR.LE.0) GO TO 90
      DO 80 J=1,ICR
      WRITE (IOUT,450) ISPNCR(J),ISPNCR(J),REDCR(J),
     1 ISPNCR(J),FALFCR(J)
   80 CONTINUE
   90 CONTINUE
      IF (CFHC.GT.0..OR.CFNOX.GT.0.) WRITE (IOUT,420) CFHC,CFNOX,
     1CHCNOX
      IF (X(2).GT.0.) WRITE (IOUT,430) ROZ2,RNMOC2,FXNS,PLSP(1)
      IF (X(2).LT.0.) WRITE (IOUT,440) ROZ2,RNMOC2,FXNS,ALREHC,ALRENX,
     1 PLSP(1)
C
C  IF BASE YEAR PRECURSOR TRANSPORT IS LESS THAN BACKGROUND
C   THEN SKIP APPENDIX B PROCEDURES
C
      IF (X(2).LT.0.) GO TO 270
      IF (HCIN .LE. HCBK) GO TO 150
      IF (CFHC.LE.0..AND.CFNOX.LE.0.) GO TO 150
C
C  DO APPENDIX B (STEP 3)
C
      HCIN = 0.0
      DO 110 I=1, NTRYS
         CALL SIM (HC2,OXN2,ZN2,10)
         OZV(I)=ZN2
         HCTMP(I)=HC2
         ANXTMP(I)=HC2
         DIFF = ABS(ZN2-DOZ)/DOZ
         IF (DIFF .LE. ACCU) GO TO 120
         IF (HC1 .NE. (-99.0)) GO TO 100
      HC1=HC2
      ZN1=ZN2
      HC2=HC1*DOZ/ZN2
      OXN2=HC2/HCTNOX
      GO TO 110
  100 B=(DOZ-ZN1)*(HC2-HC1)/(ZN2-ZN1)+HC1
C
C     CHECK FOR REVERSING TREND. FIX SO WE DO NOT LOOP
C
      IF (ZN2.GT.ZN1 .AND. HC2.LT.HC1 .AND. ZN2.GT.DOZ)
     1    B = DOZ*HC2/ZN2
      IF (B.LT.0.) B=HC2/2.0
      HC1=HC2
      ZN1=ZN2
      HC2=B
      OXN2=HC2/HCTNOX
  110 CONTINUE
      IFL=1
      GO TO 180
```

```
C
C  DO STEPS 4, 5, AND 6 (APPENDIX B)
C
  120 HCIN = HC2*CFHC
      XNIN = OXN2*CFNOX
      HCTNOX = CHCNOX
      HC1=-99.
      RNMOC2 = (HCIN - BNMOC)*RNMOC + BNMOC
      HC2 = OXN2*HCTNOX
      IF (CFHC.LE.0..AND.CFNOX.LE.0.) GO TO 150
      DO 140 J=1,ISP
      IF (EMSP(J).NE.IIHC.AND.EMSP(J).NE.IINX) GO TO 140
      IF (EMSP(J).EQ.IIHC) CFX=CFHC
      IF (EMSP(J).EQ.IINX) CFX=CFNOX
      DO 130 I=1,IEM
      EM(I)=EMO(I,J)/(1.-CFX)
  130 CONTINUE
      CALL EMISS(IEM,EM,ECO(1,J))
  140 CONTINUE
C
C  NOW DO BASE CASE CALCULATION
C
  150 DO 170 I=1, NTRYS
          CALL SIM (HC2,OXN2,ZN2,10)
          OZV(I)=ZN2
          HCTMP(I)=HC2
          ANXTMP(I)=HC2
          DIFF = ABS(ZN2-DOZ)/DOZ
          IF (DIFF .LE. ACCU) GO TO 270
          IF (HC1 .NE. (-99.0)) GO TO 160
      HC1=HC2
      ZN1=ZN2
      HC2=HC1*DOZ/ZN2
      OXN2=HC2/HCTNOX
      GO TO 170
  160 B=(DOZ-ZN1)*(HC2-HC1)/(ZN2-ZN1)+HC1
C
C     CHECK FOR REVERSING TREND. FIX SO WE DO NOT LOOP
C
      IF (ZN2.GT.ZN1 .AND. HC2.LT.HC1 .AND. ZN2.GT.DOZ)
     1    B = DOZ*HC2/ZN2
      IF (B.LT.0.) B=HC2/2.0
      HC1=HC2
      ZN1=ZN2
      HC2=B
      OXN2=HC2/HCTNOX
  170 CONTINUE
      IF (IFL.EQ.3) GO TO 180
      IFL=2
C
```

```
C     WAS NOT ABLE TO FIND DESIGN OZONE WITHIN ALLOTTED TRYS
C
  180 IF (IFL.LT.3) WRITE (IOUT,460) DOZ,NTRYS
      IF (IFL.EQ.3) WRITE (IOUT,550) DOZ
      IF (IFL.EQ.3) GO TO 330
C
C  CHECK TO SEE IF THERE IS A MAXIMUM ALONG THIS HC/NOX RATIO.
C  IF SO, DETERMINE IF DESIGN OZONE OCCURS ON LINE AND IF THERE ARE TWO
C  LOCATIONS.  IF THERE ARE TWO LOCATIONS THEN USE THE LOCATION CLOSEST
C  TO THE MEASURED NMOC AND NOX VALUES.
C
C  FIRST SORT VALUES
C
      HMIN1=999.
      DO 200 J=1,NTRYS
      DO 190 I=1,NTRYS
      IF (HCTMP(I).LT.HMIN1) INOW=I
      IF (HCTMP(I).LT.HMIN1) HMIN1=HCTMP(I)
  190 CONTINUE
      HCSORT(J)=HCTMP(INOW)
      ANSORT(J)=ANXTMP(INOW)
      OZSORT(J)=OZV(INOW)
      HCTMP(INOW)=999.
      HMIN1=999.
  200 CONTINUE
C
C  NOW FIND MAXIMUM VALUE
C
      CALL EDGMX (HCSORT,OZSORT,NTRYS,HMX,OZHX,LHX)
      MM=LHX+1
      IF (MM.LT.NTRYS.AND.OZHX.LT.DOZ) WRITE (IOUT,530) DOZ,OZHX
      IF (MM.LT.NTRYS.AND.OZHX.LT.DOZ) GO TO 330
      IF (MM.GE.NTRYS) GO TO 330
C
C  THERE IS A MAXIMUM ALONG NMOC/NOX
C
C  FIND THE LOCATIONS
C
      DO 210 J=1,LHX
      HCTMP(J)=HCSORT(J)
      OZV(J)=OZSORT(J)
  210 CONTINUE
      HCTMP(MM)=HMX
      OZV(MM)=OZHX
      SX=-SIG*3.0
      IF (MM.GT.2) SX=-SIG*60./(FLOAT(MM-2)**2)
  220 CALL CURV1(MM,OZV,HCTMP,SP1,SP2,YP,TM,SX)
      IT=1
      HCML1=CURV2(DOZ,MM,OZV,HCTMP,YP,SX,IT)
      IF (IT.GT.0) GO TO 230
      SX=SX*10.
      SX=AMAX1(-50.,SX)
      GO TO 220
  230 CONTINUE
C
C NOW DO UPPER PORTION
C
      MDIF=NTRYS-LHX
      DO 240 J=1,MDIF
      MJ=NTRYS-J
      OZV(J)=OZSORT(MJ)
      HCTMP(J)=HCSORT(MJ)
  240 CONTINUE
      OZV(MDIF+1)=OZHX
      HCTMP(MDIF+1)=HMX
      MDIF=MDIF+1
      SX=-SIG*3.0
      IF (MDIF.GT.2) SX=-SIG*60./(FLOAT(MDIF-2)**2)
  250 CALL CURV1(MDIF,OZV,HCTMP,SP1,SP2,YP,TM,SX)
      IT=1
      HCML2=CURV2(DOZ,MDIF,OZV,HCTMP,YP,SX,IT)
      IF (IT.GT.0) GO TO 260
      SX=SX*10.
      SX=AMAX1(-50.,SX)
      GO TO 250
  260 CONTINUE
C
C  SEE WHICH VALUE IS CLOSEST TO MEASURED NMOC AND NOX
C
      XNM1=HCML1/HCTNOX
      XNM2=HCML2/HCTNOX
      HCD1=SQRT((HCML1-HCMEAS)**2 + (XNM1-ANMEAS)**2)
      HCD2=SQRT((HCML2-HCMEAS)**2 + (XNM2-ANMEAS)**2)
      WRITE (IOUT,540) HCML1,XNM1,HCML2,XNM2,HCMEAS,ANMEAS,PLSP(1)
      HC2=HCML1
      OXN2=XNM1
      IF (HCD2.LT.HCD1) HC2=HCML2
      IF (HCD2.LT.HCD1) OXN2=XNM2
      IF (HCML1.LT.HCSORT(1).OR.HCML1.GT.HCSORT(NTRYS)) HC2=HCML2
      IF (HCML1.LT.HCSORT(1).OR.HCML1.GT.HCSORT(NTRYS)) OXN2=XNM2
      IF (HCML2.LT.HCSORT(1).OR.HCML2.GT.HCSORT(NTRYS)) HC2=HCML1
      IF (HCML2.LT.HCSORT(1).OR.HCML2.GT.HCSORT(NTRYS)) OXN2=XNM1
      IF (HC2.LT.HCSORT(1).OR.HC2.GT.HCSORT(NTRYS)) WRITE(IOUT,570)
      IF (HC2.LT.HCSORT(1).OR.HC2.GT.HCSORT(NTRYS)) GO TO 330
      IF (IFL.EQ.1) GO TO 120
      IFL=3
      GO TO 150
C
C     FOUND IT.   NOW DO FUTURE YEAR CALCULATION
C
```

```
  270 IF (X(2).LT.0.) OXN2=ALRENX
      IF (X(2).LT.0.) HC2=ALREHC
      OXN1 = OXN2*REDNOX
      HCORG = HC2
      IF (X(6).LT.(-0.000001)) GO TO 330
      HCAL=RNMOC1
      HCIN=RNMOC2
      OZAL=ROZ1
      OZIN=ROZ2
      IF (FOCS.GT.(-0.00001)) HCIN=FOCS
      IF (FXNS.GE.(-0.00001)) XNIN=FXNS
      IF (FXNA.GE.(-0.00001)) XNAL=FXNA
C
C  ALLOW FOR OTHER EMISSIONS CREDITS (ONLY FOR CO CURRENTLY)
C
      IF (ICR.LE.0) GO TO 290
      DO 280 J=1,ICR
      SPCR69(J)=SPCR69(J)*(1.+REDCR(J)*0.01)
      SURFCR(J)=FSRFCR(J)
      ALOFCR(J)=FALFCR(J)
  280 CONTINUE
  290 CONTINUE
C
C  TAKE AN INITIAL GUESS AT VOC REDUCTION (50 PERCENT)
C
      HC1=0.5*HC2
      HC2 = 0.0
      ZN2 = 0.0
      OXN2 = OXN1
      HC3 = 0.5
      DO 310 I=1, NTRYS
         II=I
         CALL SIM (HC1,OXN1,ZN1,10)
         DIFF = ABS(ZN1-AOZ)/AOZ
         IF (DIFF .LE. ACCU) GO TO 320
         IF (ZN1.GT.AOZ.AND.HC3.LE.0.) II=NTRYS+1
         IF (ZN1.GT.AOZ.AND.HC3.LE.0.) GO TO 320
         DELTA = 1.0
         IF (ZN1 .NE. ZN2) DELTA = (HC1 - HC2)/(ZN1 - ZN2)
         IF ((ABS(AOZ-ZN1) .GT. ABS(AOZ-ZN2)) .AND. I .NE. 1) GO TO 300
         HC3 = HC1 + (AOZ - ZN1)*DELTA
         ZN2 = ZN1
         HC2 = HC1
         HC1 = HC3
         IF (HC1 .LT. 0.0) HC1 = 0.0
         GO TO 310
  300    HC1 = HC2 + (AOZ - ZN2)*DELTA
         HC3 = HC1
         IF (HC1 .LT. 0.0) HC1 = 0.0
  310 CONTINUE
```

```
C
  320 IF (II.GE.NTRYS) WRITE (IOUT,460) AOZ,I
      IF (II.GE.NTRYS) GO TO 330
      REDUC = (HCORG - HC1)*100.0/HCORG
      IF (REDUC.LE.90.) WRITE (IOUT,470) REDUC
      IF (REDUC.GT.90.) WRITE (IOUT,560) REDUC
C
C  DO SINGLE CALC AT THIS POINT BASED ON USER INPUT OF VOC CHANGE
C
  330 IF (ABS(X(6)).EQ.0.) GO TO 340
      HC1=HCORG*(1.+HCLV*0.01)
      WRITE (IOUT,480) HCLV,PLSP(1)
      CALL SIM(HC1,OXN1,ZN1,10)
C
C  DO OZIPM-2 EKMA OPTION TO GENERATE CHANGE IN OZONE
C
  340 IF (X(5).LE.0.) GO TO 360
C
C  THE FOLLOWING ALGORITHM WAS DEVELOPED BY EPA (GIPSON, 1984)
C
      WRITE(IOUT,490)
      WRITE(IOUT,500)
      DELHC=10.0
      DO 350 K=1,11
      DELHC=DELHC-10.
      HC2=HCORG*(1.0+DELHC/100.)
      CALL SIM(HC2,OXN1,ZN,5)
      DELO3=((ZN-DOZ)/DOZ)*100.
      WRITE(IOUT,510) HC2,OXN1,ZN,DELHC,X(3),DELO3
      OZOUT(K)=ZN
  350 CONTINUE
      IF (X(5).EQ.2) WRITE(IOZC,520) OZOUT
C
C  RESET VALUES FOR NEXT OPTION
C
  360 CONTINUE
C
C  RESET EMISSIONS IF APPENDIX B WAS USED
C
      IF (HCINSV.LE.HCBK.AND.XNIN.LE.XNBK) GO TO 380
      IF (CFHC.LE.0..AND.CFNOX.LE.0.) GO TO 380
      DO 370 J=1,ISP
      CALL EMISS(IEM,EMO(1,J),ECO(1,J))
  370 CONTINUE
  380 CONTINUE
      NPTO=0
      INFO=0
      OZIN=OZINSV
      OZAL=OZALSV
      HCIN=HCINSV
      HCAL=HCALSV
      XNIN=XNINSV
      XNAL=XNALSV
      IF (ICR.LE.0) RETURN
      DO 390 J=1,ICR
      SPCR69(J)=SPCRSV(J)
      SURFCR(J)=SCRSV(J)
      ALOFCR(J)=ALCRSV(J)
  390 CONTINUE
C
      RETURN
C
C  FORMAT STATEMENTS
C
  400 FORMAT (7F10.5)
  410 FORMAT (1H1,  // 40X,43HEKMA CALCULATIONS ARE PERFORMED TO ESTIMAT
     1E / 1H0,39X,   38HSITE-SPECIFIC VOC CONTROL REQUIREMENTS // 1H0
     2,39X,  36HBASE YEAR OZONE                      ,F10.3,4H PPM / 1H0,
     339X ,36HBASE YEAR NMOC/NOX                     ,F10.3 / 1H0,39X,
     4  36HANTICIPATED CHANGE IN NOX               ,F10.3,8H PERCENT / 1H0,
     539X, 36HFUTURE OZONE TRANSPORTED ALOFT         ,F10.3,4H PPM / 1H0,
     639X, 36HFUTURE NMOC TRANSPORTED ALOFT         ,F10.3,5H PPMC / 1H0,
     739X,  36HFUTURE NOX TRANSPORTED ALOFT          ,F10.3,5H PPM  )
  420 FORMAT (1H0,39X,
     1  36HMEDIAN CONTRIBUTION FACTOR FOR NMOC ,F10.3 / 1H0,39X,
     2  36HMEDIAN CONTRIBUTION FACTOR FOR NOX  ,F10.3 / 1H0,39X,
     3  36HADJUSTED NMOC/NOX                   ,F10.3)
  430 FORMAT (1H0,39X,
     1     36HFUTURE OZONE IN THE SURFACE LAYER     ,F10.3,4H PPM / 1H0,
     239X, 36HFUTURE NMOC IN THE SURFACE LAYER     ,F10.3,5H PPMC / 1H0,
     339X,  36HFUTURE NOX IN THE SURFACE LAYER     ,F10.3,5H PPM  ,
     4 ////1H1,///,40X,36HTHE FOLLOWING SIMULATIONS WERE DONE./1H0,
     5    15X,4HNMOC,17X,3HNOX,15X,5HRATIO,17X,A4,16X,4HTIME)
  440 FORMAT (1H0,39X,
     1     36HFUTURE OZONE IN THE SURFACE LAYER     ,F10.3,4H PPM / 1H0,
     239X, 36HFUTURE NMOC IN THE SURFACE LAYER     ,F10.3,5H PPMC / 1H0,
     339X, 36HFUTURE NOX IN THE SURFACE LAYER      ,F10.3,5H PPM  / 1H0,
     439X, 55HBASE YEAR NMOC AND NOX LOCATIONS ARE ALREADY DETERMINED./,
     51H0,
     639X, 36HBASE YEAR INITIAL NMOC LOCATION      ,F10.3,5H PPMC / 1H0,
     739X,  36HBASE YEAR NOX LOCATION              ,F10.3,5H PPM  ,
     8 ////1H1,///40X,36HTHE FOLLOWING SIMULATIONS WERE DONE./1H0,
     9    15X,4HNMOC,17X,3HNOX,15X,5HRATIO,17X,A4,16X,4HTIME)
C
  450 FORMAT (1H0,39X,37HEMISSION CREDITS WILL BE ALLOWED FOR ,A4,/,1H0,
     1 39X,22HANTICIPATED CHANGE IN ,A4,10X,F10.3,8H PERCENT ,  /,1H0,
     239X,7HFUTURE ,A4,25H TRANSPORTED ALOFT        ,F10.3,4H PPM  )
C
  460 FORMAT (1H0 // 40X,15HCOULD NOT FIND ,F10.4,10H OZONE IN ,
     1       I3,6H TRYS ,//,40X,34HCHECK INPUT CONDITIONS OR GENERATE
  2        ,/,40X,20HAN ISOPLETH DIAGRAM.  )
470 FORMAT (1H0 // 40X,27HVOC CONTROL REQUIREMENT IS ,F5.1,8H PERCENT)
480 FORMAT (1H1 // 40X,41HTHE FOLLOWING SIMULATION WAS DONE WITH A
   1 / 1H0,39X,F7.2,24H PERCENT CHANGE IN NMOC.    /1H0,
   2     15X,4HNMOC,17X,3HNOX,15X,5HRATIO,17X,A4,16X,4HTIME)
490 FORMAT(1H1,39X,'EKMA PREDICTED CHANGES IN OZONE')
500 FORMAT(///40X,'HC',6X,'NOX',6X,'O3',4X,'%-CHG HC',3X,
   1'%-CHG NOX',3X,'%-CHG O3'/)
510 FORMAT(38X,F6.3,3X,F5.3,3X,F5.4,4X,F6.1,5X,F6.1,
   16X,F6.1)
520 FORMAT(11F5.3)
530 FORMAT (1H0,39X, 23HTHE BASE YEAR OZONE OF ,F10.3,43H PPM DOES NOT
   1 OCCUR WITH THE GIVEN INPUTS. ,/,40X, 32HTHE MAXIMUM CALCULATED OZ
   2ONE IS ,F10.3,5H PPM.//)
540 FORMAT (1H0,39X,44HTHE BASE YEAR OZONE OCCURS AT TWO LOCATIONS ,/,
   140X,25HALONG THE NMOC/NOX RATIO:,/,1H0,
   2    39X,4HNMOC,17X,3HNOX,/,35X,F11.5,7X,F11.5,/,35X,F11.5,7X,
   3    F11.5,//,40X,45HTHE LOCATION CLOSEST TO THE MEASURED NMOC OF ,
   4 F10.3, 5H PPMC ,/,40X,11HAND NOX OF ,F10.3,18H PPM WILL BE USED.
   5, / , 40X,50HTHE FOLLOWING SIMULATIONS WERE DONE AT THIS POINT.,/,
   61H0,14X,4HNMOC,17X,3HNOX,15X,5HRATIO,17X,A4,16X,4HTIME)
550 FORMAT (1H0,39X,21HCANNOT FIND OZONE OF ,F5.2,17H AFTER SECOND SET
   1,/,40X,56HOF ITERATIONS.  CHECK INPUTS OR GENERATE AN ISOPLETH DIA
   25HGRAM. )
560 FORMAT (1H0,39X,33HNOTE THE CALCULATED REDUCTION OF ,F6.2,8H PERCE
   1NT,/,40X,54HMAYBE UNREALISTIC WITH THE GIVEN INPUTS.
   2,/,40X,35HAN ISOPLETH DIAGRAM IS RECOMMENDED. )
570 FORMAT (1H0,39X,44HBOTH CALCULATED NMOC AND NOX LOCATIONS ARE  ,/,
   140X,57HOUTSIDE RANGE OF INTERPOLATION.  CHECK INPUTS OR GENERATE
   2,/,40X,20HAN ISOPLETH DIAGRAM.  )
C
      END
      SUBROUTINE SIM (HCHH,XNHH,ZN,INX)
C
C     THIS ROUTINE CONTROLS THE SIMULATION OF AN INDIVIDUAL CALCULATION
C
      SAVE
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /TITL/ ITTL(36)
      COMMON /HOUR/ OZM(5),NGO,TTM,TM(5)
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /EMIS/ NEM,IS1,ESTRT(5),ESTOP,ESLP,EMSP(5),IEMLS(5),EOSLP(5
     1),EMO(26,5),ECI(5),EM(26),EC(125),ECO(125,5)
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,II03,
     1IIH2O,IBZA,JPLUS
      COMMON /FRPLOT/ SAVCON(80,5),SAVTIM(80),NT,INOW
      COMMON /STORE/ AST(60)
      COMMON /INOUT/ INU,IOUT,ITAPE,IALN,IALL,IN1,IOZC
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),CI(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),KK(24),JSTRT,JSTOP,SPEE,MNLM,MXLM,MAXL,MAXJ
      COMMON /HJH/ HCSAV,XNSAV
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      COMMON /MIX/ NMIX,AMIX(26),STRM,STOPM,DC(104)
      COMMON /MIXING/ DSTRT,DEND,AMC(5),BMC(5),CMC(5),FD(6),FG(6),
     1AMXX(26),DL,TTMAX,SSRISE,SRMIN,DELH,TDIL,NMXX,HEIGHT,SSET,SRISE
      COMMON /BK1/ FBK(10),FBKAL(5),HCBK,XNBK,OZBK,H2OBK
      COMMON /CRED/ ICR,ISPCR,ISPNCR(3),SPCR69(3),SURFCR(3),ALOFCR(3),
     1REDCR(3),FSRFCR(3),FALFCR(3),COSFBK,COAFBK
      DIMENSION RT(135), RX(20), C(61), CO3(5), INST(3,5)
      INTEGER REACT,SPECIS,HCSPEC,PLSP,SPEE,EMSP
      DATA INST/'(INS','TANT',')    ','(INS','TANT',')    ' ,
     1'(INS','TANT',')    ' ,'(INS','TANT',')    ' ,'(INS','TANT',')    ' /
      ISP=IS1+ICR
      TCI=FLOAT(JSTRT)
      HC=HCHH
      XN=XNHH
      START=0.
      T=START
      TDIL=0.
      TTRS=CLKMIN (TCI)
      TTR1=CLKMIN (SRISE)
      TTR2=CLKMIN (SSET)
      TR1=TTR1-TTRS
      TR2=TTR2-TTRS
      IF (TR2.LE.0.) TR2=2000.
      IF (TR1.GE.0..OR.((JSTOP-JSTRT).GT.100)) GO TO 10
      TR2=1440.+TR1
      TR1=TTR2-TTRS
      IF (TR1.LT.0.) TR1=0.
      IF (TR2.LE.0.) TR2=2000.
   10 IF (IN1.EQ.1) SAV=STOPP
      STOPP=SAV
      FLST=1.0
      TLST=START
      EHC=HC
      EXN=XN
      HCSAV=HC
      XNSAV=XN
      TDC=STOPP/80.
      TD=0.
      DO 20 I=2,80
      TD=TD+TDC
   20 SAVTIM(I)=TD
      SAVTIM(1)=0.
      IF (NPTO.EQ.0) TPRNT=STOPP
      IF (NPTO.NE.0) TTM=5.0
      H=1.E-10
      N=NI
      M=NS-1
      DO 30 J=1,5
   30 ECI(J)=1.
      IF (ISP.LE.2) GO TO 60
      DO 50 I=3,ISP
      DO 40 J=1,M
      IF (EMSP(J).EQ.IIHC.OR.EMSP(J).EQ.IINX) GO TO 40
      IF (SPECIS(J).NE.EMSP(I)) GO TO 40
      ECI(I)=SPRSE(I)
      IF (ICR.GT.0.AND.I.GT.2) ECI(I)=SPCR69(I-2)
      GO TO 50
   40 CONTINUE
   50 CONTINUE
   60 IFH20=0
      IFCO=0
      IF (N.LE.0) GO TO 80
      DO 70 I=1,N
      C(I)=CI(I)
      IF (REACT(I).EQ.IIH20) IFH20=1
      IF (REACT(I).EQ.IICO) IFCO=1
   70 CONTINUE
   80 CONTINUE
      IF (IFH20.NE.0) GO TO 90
      N=N+1
      IF (N.LE.0) N=1
      REACT(N)=IIH20
      C(N)=H2OBK                                                        V   101
   90 IF (IFCO.NE.0.OR.ICR.GT.0) GO TO 100                              V   102
      N=N+1                                                             V   103
      IF (N.LE.0) N=1                                                   V   104
      REACT(N)=IICO                                                     V   105
      C(N)=COSFBK                                                       V   106
  100 CALL RATES (C,N)                                                  V   107
      IF (IN1.NE.0) CALL SPARS (IA,JA,M)                                V   108
      DHC=HCIN-HCBK                                                     V   109
      DXN=XNIN-XNBK                                                     V   110
      IF (DHC.LT.0.) HCIN=HCBK                                          V   111
      IF (DXN.LT.0.) XNIN=XNBK                                          V   112
      DO 110 J=1,NHC                                                    V   113
      K=IH(J)                                                           V   114
      FBKS=FINHC(J)                                                     V   115
      IF (FBK(J).GT.(-0.00001)) FBKS=FBK(J)                             V   116
      C(K)=(HC*RCTY(J)+HCIN*FINHC(J))/CARB(J)                           V   117
  110 CONTINUE                                                          V   118
      DO 120 K=1,2                                                      V   119
      I=INOX(K)                                                         V   120
      IF (SPECIS(I).EQ.IINO2) C(I)=XN*XNF(K)+XNIN                       V   121
      IF (SPECIS(I).NE.IINO2) C(I)=XN*XNF(K)                            V   122
  120 CONTINUE                                                          V   123
      IF (OZIN.EQ.0.) GO TO 160                                         V   124
      DO 130 I=1,NOZ                                                    V   125
      IF (PLSP(I).NE.II03  ) GO TO 130                                  V   126
      JOZ=KOZ(I)                                                        V   127
      C(JOZ)=OZIN                                                       V   128
      GO TO 160                                                         V   129
  130 CONTINUE                                                          V   130
  140 DO 150 I=1,N                                                      V   131
      IF (SPECIS(I).NE.II03  ) GO TO 150                                V   132
      C(I)=OZIN                                                         V   133
      GO TO 160                                                         V   134
  150 CONTINUE                                                          V   135
  160 IF (ICR.LE.0) GO TO 190                                           V   136
      DO 180 J=1,ICR                                                    V   137
      DO 170 I=1,N                                                      V   138
      IF (SPECIS(I).NE.ISPNCR(J)) GO TO 170                             V   139
      C(I)=SURFCR(J)+SPCR69(J)                                          V   140
      GO TO 180                                                         V   141
  170 CONTINUE                                                          V   142
  180 CONTINUE                                                          V   143
  190 IF (ISP.LE.2.OR.ICR.GT.0) GO TO 195                               V   143A
      DO 194 J=3,ISP                                                    V   143B
      DO 193 I=1,N                                                      V   143C
      IF (SPECIS(I).NE.EMSP(J)) GO TO 193                               V   143D
      IF (SPECIS(I).NE.IICO) GO TO 192                                  V   143E
      IF (IFCO.NE.0) C(I)=C(I)+SPRSE(J)                                 V   143F
      IF (IFCO.EQ.0) C(I)=SPRSE(J)                                      V   143G
      GO TO 194
  192 C(I)=C(I)+SPRSE(J)
      GO TO 194
  193 CONTINUE
  194 CONTINUE
  195 CALL DIFFUN (N,START,C,RT,2,START)
      HC=HCSAV
      XN=XNSAV
      IF (NPTO.EQ.0) GO TO 210
      WRITE (IOUT,520)
      WRITE (IOUT,540) (ITTL(I),I=1,36)
      IF (INFO.LT.1) WRITE (IOUT,600) (PLSP(I),I=1,NOZ)
      IF (INFO.LT.1) WRITE (IOUT,610) ((INST(I,KI),I=1,3),KI=1,NOZ)
      IF (INFO.LT.1) GO TO 210
      IF (NI.NE.0) WRITE (IOUT,630) (REACT(I),I=1,NI)
      IF (NI.NE.0) WRITE (IOUT,640) (CI(I),I=1,NI)
      WRITE (IOUT,560) ERR,TEMP,HEIGHT
      WRITE (IOUT,570) (R(IR),IR=1,NR)
      IF (IP.EQ.0) GO TO 210
      DO 200 I=1,IP
      K=IPH(I)
  200 RX(I)=R(K)
      WRITE (IOUT,580) (IPH(IR),IR=1,IP)
      WRITE (IOUT,590) (RX(IR),IR=1,IP)
  210 IN=1
      IF (IN1.EQ.0) IN=4
      IN1=0
      DILUT=DCON
      IF (TSRT.LE.START.AND.DTIM.NE.0.) DILUT=ALOG(Z2/Z1)/DTIM
      ERRSAV=ERR
      TNEXT=1.
      IF (NPTO.EQ.0) GO TO 310
      IF (T-START) 390,390,310
  220 IF (AMIX(1).GT.(-1.).AND.T.GE.TDIL) TDIL=TDIL+60.
      TTIM=TR1
      IF (T.GE.TTIM) TTIM=TR2
      IF (T.GE.TR2) TTIM=2000.
      IF (DTIM.EQ.0.) GO TO 300
      IF (T.GE.TSRT) GO TO 250
      DILUT=DCON
      IF (TSRT-TPRNT-20.) 230,230,240
  230 TNEXT=AMIN1(TSRT,TTIM)
      IN=2
      GO TO 310
  240 TNEXT=AMIN1(TPRNT,TTIM)
      IN=0
      GO TO 310
  250 IF (AMIX(1).LT.(-1.)) GO TO 260
      TNEXT=AMIN1(TDIL,TPRNT,TTIM)
      IF (T.GE.TSRT+DTIM) GO TO 300
      GO TO 270
  260 IF (T.GE.TSRT+DTIM) GO TO 300
      DILUT=ALOG(Z2/Z1)/DTIM
      FLST=EXP(-DCON*TSRT)
      TLST=TSRT
      TNEXT=TSRT+DTIM
      TNEXT=AMIN1(TNEXT,TTIM)
  270 IF (TNEXT-TPRNT-20.) 280,280,290
  280 IN=2
      GO TO 310
  290 TNEXT=AMIN1(TPRNT,TTIM)
      IN=0
      GO TO 310
  300 TNEXT=AMIN1(TPRNT,TTIM,STOPP)
      FLST=EXP(-DCON*TSRT)*Z1/Z2
      TLST=TSRT+DTIM
      DILUT=DCON
  310 IF (ERR.NE.ERRSAV) IN=-1
      ERR=ERRSAV
  320 CALL DRIVES (M,T,H,C,TNEXT,ERR,21,IN,IA,JA)
      T=TNEXT
      IF (NPTO.NE.0) GO TO 350
      IF (T.LT.STOPP) GO TO 220
      IF (INX.GT.(-1)) GO TO 350
      IF (INX.EQ.(-1).AND.OZM(1).GT.0.75*ZN) GO TO 350
      IF (INX.NE.(-3)) GO TO 330
      TOL=1.0-(0.17/(SQRT(FLOAT(ITAPE))))
      IF (OZM(1).GE.TOL*ZN.AND.OZM(1).LE.ZN/TOL) GO TO 350
  330 CONTINUE
      IF (INX.NE.(-2)) GO TO 340
      IF (OZM(1).LT.1.1*ZN.AND.OZM(1).GT.0.9*ZN) GO TO 350
  340 ZN=OZM(1)
      RETURN
  350 ZN=OZM(1)
      IF (NPTO.NE.0) GO TO 380
      IF (INX.LT.5) INX=1
      IF (T.LT.STOPP) GO TO 220
      IF (XN.EQ.0.) RR=0.
      IF (XN.NE.0.) RR=HC/XN
      IF (INFO.LT.0) GO TO 360
      IF (TM(1).GT.0.) WRITE (IOUT,470) HC,XN,RR,(OZM(J),J=1,NOZ)
      IF (TM(1).LE.0.) WRITE (IOUT,510) HC,XN,RR,(OZM(J),J=1,NOZ)
      WRITE (IALN,680) HC,XN,(OZM(J),J=1,NOZ)
      RETURN
  360 CONTINUE
      CTEND=CLOCK(TCI,IFIX(STOPP))
      IF (CTEND.GT.2400.) CTEND=CTEND-2400.
      DO 370 J=1,NOZ
      IF (TM(J).LE.0.) RT(J)=CTEND
      IF (TM(J).GT.0.) RT(J)=CLOCK(TCI,IFIX(TM(J)))
      IF (RT(J).GT.2400.) RT(J)=RT(J)-2400.
  370 CONTINUE
      IF (INX.LT.5)
     1WRITE (IOUT,550) HC,XN,RR,(OZM(IIC),RT(IIC),IIC=1,NOZ)
      IF (TM(1).GT.0..AND.INX.GT.5) WRITE (IOUT,660) HC,XN,RR,OZM(1)
     1,RT(1)
      IF (TM(1).LE.0..AND.INX.GT.5) WRITE (IOUT,670) HC,XN,RR,OZM(1)
      IF (INX.LT.5) WRITE (IALN,680) HC,XN,(OZM(J),J=1,NOZ)
      RETURN
  380 IF (IN.NE.0) RETURN
      IF (T.GE.STOPP) TPRNT=STOPP
      IF (T.LT.TPRNT) GO TO 220
  390 HCC=0.
      DO 400 I=1,NHC
      K=IH(I)
  400 HCC=HCC+CARB(I)*C(K)
      CNOX=0.
      DO 410 I=1,2
      K=INOX(I)
      IF (I.EQ.1) FNO2=C(K)
  410 CNOX=CNOX+C(K)
      IF (CNOX.LE.0.0000001) FNO2=0.
      IF (CNOX.GT.0.0000001) FNO2=FNO2/CNOX
      IF (CNOX.LE.0.000001) RNX=0.
      IF (CNOX.GT.0.000001) RNX=HCC/CNOX
      DO 420 I=1,NOZ
      JOZ=KOZ(I)
      CO3(I)=C(JOZ)
  420 CONTINUE
      IF (INFO.GT.0) WRITE (IOUT,520)
      IF (INFO.GT.0) WRITE (IOUT,600) (PLSP(I),I=1,NOZ)
      IF (INFO.GT.0) WRITE (IOUT,610) ((INST(I,KI),I=1,3),KI=1,NOZ)
      IF (T.EQ.START) TIMNW=START
      IF (T.NE.START) TIMNW=TPRNT
      CTIME=CLOCK(TCI,IFIX(TIMNW))
C
C  ****** UPDATED 11/81 FOR 24 HR RUNS
C
      IF (CTIME.GT.2400.) CTIME=CTIME-2400.
      WRITE (IOUT,620) CTIME,HCC,RNX,CNOX,FNO2,(CO3(I),I=1,NOZ)
      IF (INFO.LE.0) GO TO 460
      WRITE (IOUT,500) (SPECIS(I),I=1,M)
      WRITE (IOUT,530) CTIME,(C(I),I=1,10),H,(C(I),I=11,M)
      CALL DIFFUN (N,TIMNW,C,RT,1,TIMNW)
      WRITE (IOUT,480) (RT(I),I=1,M)
      DO 440 I=1,NR
      J=KR(I,1)
      IF (J.EQ.0) RT(I)=0.
      IF (J.EQ.0) GO TO 440
      JT=ITYPE(I)
      XT=1.
      DO 430 L=1,JT
      J=KR(I,L)
      XJ=1.
      IF (J.GT.0) XJ=C(J)**ISC(I,L)
      IF (ISC(I,L).EQ.(-1)) XJ=C(J)**SC(I,L)
      XT=XT*XJ
  430 CONTINUE
      RT(I)=XT*R(I)
  440 CONTINUE
      WRITE (IOUT,490) (RT(I),I=1,NR)
      IF (IP.EQ.0) GO TO 460
      DO 450 I=1,IP
      K=IPH(I)
  450 RX(I)=R(K)
      WRITE (IOUT,590) (RX(I),I=1,IP)
      WRITE (IOUT,650) HEIGHT,TEMP,ZENI
  460 IF (TNEXT.EQ.1.) ERR=AMIN1(1.,100.*ERR)
      IF (TIMNW.EQ.START) GO TO 320
      IF (NPTO.NE.0) IN1=1
      IF (T.GE.STOPP) RETURN
      TPRNT=TPRNT+TSTEP
      GO TO 220
C
C  FORMAT STATEMENTS
C
  470 FORMAT (1H0,F11.5,4X,F11.5,5X,F11.5,5X,5(1PE12.5,3X))
  480 FORMAT (/10H NET RATES,1X,1P10E12.3/(11X,1P10E12.3))
  490 FORMAT (//1X,22HTHE REACTION RATES ARE/(1H ,1P10E13.2))
  500 FORMAT (//3X,5HTIME ,4X,10(4X,A4,4X)/1X,8HINTERVAL,3X,10(4X,A4,
     14X)/(12X,10(4X,A4,4X)))
  510 FORMAT (1H0,F11.5,4X,F11.5,5X,F11.5,5X,5(1PE12.5,3X),8H NOT MAX)
  520 FORMAT (1H1/)
  530 FORMAT (/1PE11.3,10E12.3/E11.3,10E12.3/(11X,10E12.3))
  540 FORMAT (46X,36A2)
  550 FORMAT (1H0,F11.5,F9.5,F11.5,2X,4(1PE10.4,3X,0PF5.0,2X),1PE10.4,
     13X,0PF5.0)
  560 FORMAT (24H0THE ERROR TOLERANCE IS ,1PE10.3/
     125H0THE TEMPERATURE USED IS ,E10.3/
     221H0THE MIXING HEIGHT IS ,E10.2)
  570 FORMAT (29H0THE RATE CONSTANTS USED WERE//(1H0,1P10E13.3))
  580 FORMAT (29H0THE PHOTOLYSIS REACTIONS ARE/(1H0,9I13))
  590 FORMAT (34H0THE PHOTOLYTIC RATE CONSTANTS ARE/(5X,1P9E13.3))
  600 FORMAT (1H0,10X,4HTIME,6X,4HNMOC,7X,5HNMOC/,8X,3HNOX, 9X,3HNO2,9X,
     1A4,4(8X,A4))
  610 FORMAT(10X,6H(LDT ),4X,5HTOTAL,8X,3HNOX,8X,5HTOTAL,6X,8HFRACTION,
     13X,5(3A4))
  620 FORMAT (1H0,8X,F6.0,1X,F10.5,2X,F10.5,2X,F10.5,2X,F10.5,2X,F10.5,
     15(2X,F10.5))
  630 FORMAT (1H0,22HINITIAL CONCENTRATIONS//(12X,10(4X,A4,4X)))
  640 FORMAT (/(10X,1P10E12.3))
  650 FORMAT (//30H0THE CURRENT MIXING HEIGHT IS  ,F10.2/
     130H0THE CURRENT TEMPERATURE IS   ,F10.2/
     230H0THE CURRENT ZENITH ANGLE IS  ,F10.2)
  660 FORMAT (1H0,4F20.5,F21.0)
  670 FORMAT (1H0,4F20.5,14X,7HNOT MAX)
  680 FORMAT (1X,7E13.4)
  690 FORMAT (1X,F8.5,2X,F8.5,2X,F8.5,3X,5(1PE8.5,2X))
  700 FORMAT (1X,F8 5,2X,F8.5,2X,F8.5,3X,5(1PE8.5,2X),8H NOT MAX)
  710 FORMAT (1X,F8.5,F9.5,F8.5,2X,4(1PE8.4,2X,0PF6.0,2X),1PE8.4,
     12X,0PF6.0)
  720 FORMAT (1X,4HTIME,6X,4HNMOC,7X,5HNMOC/,8X,3HNOX, 9X,3HNO2,9X,
     1A4,4(8X,A4))
  730 FORMAT(6H(LDT ),4X,5HTOTAL,8X,3HNOX,8X,5HTOTAL,6X,8HFRACTION,
     13X,5(3A4))
  740 FORMAT (1X,F6.0,1X,F10.5,2X,F10.5,2X,F10.5,2X,F10.5,2X,F10.5,
     15(2X,F10.5))
  750 FORMAT (1X,4F10.5,F11.0)
  760 FORMAT (1X,4F10.5,4X,7HNOT MAX)
      END
      SUBROUTINE DRIVES (N,T0,H0,Y0,TOUT,EPS,MF,INDEX,IA,JA)
C
C     THIS IS THE DRIVER ROUTINE FOR THE GEAR INTEGRATION SCHEME
C
C     REF: SPELLMANN AND HINDMARSH (1975)
C
      SAVE
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /HJH/ HCSAV,XNSAV
      DIMENSION IA(1), JA(1), Y0(N)
      DIMENSION Y(61,6)
      COMMON /GEAR1/ T,H,HMIN,HMAX,EPSC,UROUND,NC,MFC,KFLAG,JSTART
      COMMON /GEAR2/ YMAX(100)/GEAR3/ERROR(100)/GEAR4/W1(60,3)
      COMMON /GEAR5/ IW1(61,9)/GEAR6/W2(2000)/GEAR7/IW2(2000)
      COMMON /GEAR8/ EPSJ,IPTI2,IPTI3,IPTI4,IPTR2,IPTR3,NGRP
      COMMON /GEAR9/ HUSED,NQUSED,NSTEP,NFE,NJE,NZA,NPL,NPU,NZL,NZU,NZRO
      COMMON /INOUT/ INP,LOUT,ITAPE,IALN,IALL,INHH,IOZC
      INTEGER HCSPEC,REACT,PLSP
      DATA NMX/60/,LENW2/2000/,LENIW2/2000/
      NGP=0
      IF (INDEX.EQ.4) GO TO 30
      IF (INDEX.EQ.0) GO TO 60
      IF (INDEX.EQ.2) GO TO 70
      IF (INDEX.EQ.(-1)) GO TO 80
      IF (INDEX.EQ.3) GO TO 90
      IF (INDEX.NE.1) GO TO 270
      IF (EPS.LE.0.) GO TO 240
      IF (N.LE.0) GO TO 250
      IF ((T0-TOUT)*H0.GE.0.) GO TO 260
      MITER=MF-10*(MF/10)
      IF ((MITER.NE.1).AND.(MITER.NE.2)) GO TO 30
      NP1=N+1
      NZA=IA(NP1)-1
      MAX=LENIW2/2
      IPTI2=MAX+1
      CALL SORDER (N,IA,JA,IW1,IW1(1,5),MAX,IW2,IW2(IPTI2),IER)
      IPTI2=NZA+1
      IF (IPTI2+NZA-1.GT.LENIW2) GO TO 290
      DO 10 I=1,NP1
   10 IW1(I,2)=IA(I)
      DO 20 I=1,NZA
   20 IW2(I)=JA(I)
      CALL NSCORD (N,IW1(1,2),IW2,IW1(1,3),IW2(IPTI2),IW1,IW1(1,5),IW1(1
     1,8))
      MAXPL=(LENIW2-NZA)/2
      IPTI3=IPTI2+MAXPL
      MAXPU=LENIW2-IPTI3+1
      CALL NSSFAC (N,IW1(1,2),IW2,MAXPL,IW1(1,3),IW2(IPTI2),IW1(1,4),MAX
     1PU,IW1(1,5),IW2(IPTI3),IW1(1,6),Y(1,6),IW1(1,9),Y,Y(1,2),Y(1,3),IW
     21(1,7),IW1(1,8),Y(1,4),Y(1,5),IER)
      NPL=IW1(N,4)
      NPU=IW1(N,6)
      NZL=IW1(N+1,3)
      NZU=IW1(N+1,5)
      IPTR2=NZA+1
      IPTR3=IPTR2+MAXO(NZA,NZL)
      IF (IPTR3+MAXO(NZA,NZU)-1.GT.LENW2) GO TO 290
   30 DO 40 I=1,N
      YMAX(I)=ABS(YO(I))
      IF (YMAX(I).EQ.0.) YMAX(I)=1.E-10
   40 Y(I,1)=YO(I)
      NC=N
      T=TO
      H=HO
      NZRO=0
      TST=EPS*1.E-10
      DO 50 I=1,N
   50 IF (Y(I,1).GT.TST) NZRO=NZRO+1
      NZRO=MAXO(NZRO,1)
      NOLD=NZRO
      HMIN=ABS(HO)
      HMAX=ABS(TO-TOUT)*10.
      HMAX=AMIN1(HMAX,20.)
      EPSC=EPS
      MFC=MF
      JSTART=0
      NO=N
      CALL OZMX (Y,TL,TOUT,NO)
      NMX1=NMX+1
      EPSJ=SQRT(UROUND)
      NHCUT=0
      GO TO 100
   60 HMAX=ABS(TOUT-TOUTP)*10.
      HMAX=AMIN1(HMAX,20.)
      GO TO 160
   70 HMAX=ABS(TOUT-TOUTP)*10.
      HMAX=AMIN1(HMAX,20.)
      IF ((T-TOUT)*H.GE.0.) GO TO 300
      GO TO 170
   80 IF ((T-TOUT)*H.GE.0.) GO TO 280
      JSTART=-1
      NC=N
      EPSC=EPS
      TST=EPS*1.E-10
      MFC=MF
   90 CONTINUE
```

```
  100 CONTINUE
      CALL STIFFS (Y,NO,IA,JA,W1,NMX,IW1,NMX1)
      JOZ=KOZ(1)
      KGO=1-KFLAG
      GO TO (110,190,220,200),KGO
  110 CONTINUE
      IF (T.GE.TL) CALL OZMX (Y,TL,TOUT,NO)
      D=0.
      NZRO=0
      DO 140 I=1,NC
      IF (Y(I,1).GE.0.) GO TO 130
      NGP=NGP+1
      DO 120 J=1,6
      K=(J-1)*N+I
  120 Y(K,1)=0.
  130 CONTINUE
      IF (Y(I,1).GT.TST) NZRO=NZRO+1
      AYI=ABS(Y(I,1))
      YMAX(I)=AMAX1(1.E-10,AYI)
  140 D=D+(AYI/YMAX(I))**2
      NZRO=MAX0(NZRO,1)
      DO 150 II=1,NC
  150 YO(II)=Y(II,1)
      CALL SAVIT (T,YO)
      IF (NZRO.NE.NOLD) JSTART=-1
      IF (NZRO.NE.NOLD) NOLD=NZRO
      D=D*(UROUND/EPS)**2
      IF (D.GT.FLOAT(N)) GO TO 230
      IF (INDEX.EQ.3) GO TO 300
      IF (INDEX.EQ.2) GO TO 170
  160 IF ((T-TOUT)*H.LT.0.) GO TO 90
      CALL INTERP (TOUT,Y,NO,YO)
      GO TO 320
  170 IF (T.GE.TOUT) GO TO 180
      IF (((T+H)-TOUT).LE.1..AND.T.GT.1.) GO TO 90
      H=(TOUT-T)*(1.+4.*UROUND)
      JSTART=-1
      GO TO 90
  180 JSTART=-1
      H=AMIN1(H,1.)
      GO TO 300
  190 CONTINUE
  200 IF (NHCUT.EQ.10) GO TO 210
      NHCUT=NHCUT+1
      HMIN=.1*HMIN
      H=.1*H
      JSTART=-1
      GO TO 90
  210 WRITE (LOUT,330)
      IF (KGO.EQ.4) WRITE (LOUT,360) T
      STOP
  220 WRITE (LOUT,340) T,H
      STOP
  230 WRITE (LOUT,350) T
      KFLAG=-2
      STOP
  240 WRITE (LOUT,370)
      STOP
  250 WRITE (LOUT,380)
      STOP
  260 WRITE (LOUT,390)
      STOP
  270 WRITE (LOUT,400) INDEX
      STOP
  280 WRITE (LOUT,410) T,TOUT,H
      STOP
  290 WRITE (LOUT,420)
      STOP
  300 TOUT=T
      DO 310 I=1,N
  310 YO(I)=Y(I,1)
      CALL SAVIT (TOUT,YO)
  320 INDEX=KFLAG
      TOUTP=TOUT
      HO=HUSED
      IF (KFLAG.NE.0) HO=H
      RETURN
C
C  FORMAT STATEMENTS
C
  330 FORMAT (//44H PROBLEM APPEARS UNSOLVABLE WITH GIVEN INPUT//)
  340 FORMAT (//35H KFLAG = -2 FROM INTEGRATOR AT T = ,E16.8,5H  H =,E16
     1.8/52H  THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED//)
  350 FORMAT (//37H INTEGRATION HALTED BY DRIVER AT T = ,E16.8/56H  EPS
     1TOO SMALL TO BE ATTAINED FOR THE MACHINE PRECISION/)
  360 FORMAT (//35H KFLAG = -3 FROM INTEGRATOR AT T = ,E16.8/45H  CORREC
     1TOR CONVERGENCE COULD NOT BE ACHIEVED/)
  370 FORMAT (//28H ILLEGAL INPUT.. EPS .LE. 0.//)
  380 FORMAT (//25H ILLEGAL INPUT.. N .LE. 0//)
  390 FORMAT (//36H ILLEGAL INPUT.. (TO-TOUT)*H .GE. 0.//)
  400 FORMAT (//24H ILLEGAL INPUT.. INDEX =,I5//)
  410 FORMAT (//44H INDEX = -1 ON INPUT WITH (T-TOUT)*H .GE. 0./4H T =,E
     116.8,9H   TOUT =,E16.8,6H   H =,E16.8)
  420 FORMAT (//42H INSUFFICIENT WORKING STORAGE IN IW2 OR W2//)
      END
      SUBROUTINE STIFFS (Y,NO,IA,JA,W1,NMX,IW1,NMX1)
C
C  THE GEAR PREDICTOR/SOLVER ROUTINE FOR STIFF EQUATIONS
C
      SAVE
      COMMON /GEAR1/ T,H,HMIN,HMAX,EPS,UROUND,N,MF,KFLAG,JSTART
      COMMON /GEAR2/ YMAX(100)/GEAR3/ERROR(100)
      COMMON /GEAR6/ W2(2000)/GEAR7/IW2(2000)
      COMMON /GEAR8/ EPSJ,IPTI2,IPTI3,IPTI4,IPTR2,IPTR3,NGRP
      COMMON /GEAR9/ HUSED,NQUSED,NSTEP,NFE,NJE,IDUMMY(5),NZRO
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IB(60)
     1,JB(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /PHOTON/ CF(72,20),PX(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      DIMENSION Y(NO,1), IA(1), JA(1), W1(NMX,1), IW1(NMX1,1)
      DIMENSION EL(13), TQ(4), RT(3)
      INTEGER SPECIS
      DATA EL(2)/1./,OLDLO/1./
      KFLAG=0
      TOLD=T
      IF (JSTART.GT.0) GO TO 100
      IF (JSTART.NE.0) GO TO 20
      CALL DIFFUN (N,T,Y,W1,1,TOLD)
      DO 10 I=1,N
   10 Y(I,2)=H*W1(I,1)
      METH=MF/10
      MITER=MF-10*METH
      NQ=1
      L=2
      IDOUB=3
      RMAX=1.E4
      RC=0.
      CRATE=1.
      HOLD=H
      MFOLD=MF
      NSTEP=0
      NSTEPJ=0
      NFE=1
      NJE=0
      IRET=3
      GO TO 30
   20 IF (MF.EQ.MFOLD) GO TO 50
      MEO=METH
      MIO=MITER
      METH=MF/10
      MITER=MF-10*METH
      MFOLD=MF
      IF (MITER.NE.MIO) IWEVAL=MITER
      IF (METH.EQ.MEO) GO TO 50
      IDOUB=L+1
      IRET=1
   30 CALL COSET (METH,NQ,EL,TQ,MAXDER)
      LMAX=MAXDER+1
      RC=RC*EL(1)/OLDLO
      OLDLO=EL(1)
   40 FN=FLOAT(NZRO)
      EDN=FN*(TQ(1)*EPS)**2
      E=FN*(TQ(2)*EPS)**2
      EUP=FN*(TQ(3)*EPS)**2
      BND=FN*(TQ(4)*EPS)**2
      EPSOLD=EPS
      NOLD=NZRO
      GO TO (60,70,100),IRET
   50 IF (EPS.EQ.EPSOLD.AND.NZRO.EQ.NOLD) GO TO 60
      IRET=1
      GO TO 40
   60 IF (H.EQ.HOLD) GO TO 100
      RH=H/HOLD
      H=HOLD
      IREDO=3
      GO TO 80
   70 RH=AMAX1(RH,HMIN/ABS(H))
   80 RH=AMIN1(RH,HMAX/ABS(H),RMAX)
      R1=1.
      DO 90 J=2,L
      R1=R1*RH
      DO 90 I=1,N
   90 Y(I,J)=Y(I,J)*R1
      H=H*RH
      RC=RC*RH
      IDOUB=L+1
      IF (IREDO.EQ.0) GO TC 550
  100 IF (ABS(RC-1.).GT.0.3) IWEVAL=MITER
      IF (NSTEP.GE.NSTEPJ+20) IWEVAL=MITER
      T=T+H
      DO 110 J1=1,NQ
      DO 110 J2=J1,NQ
      J=(NQ+J1)-J2
      DO 110 I=1,N
  110 Y(I,J)=Y(I,J)+Y(I,J+1)
  120 DO 130 I=1,N
      IF (Y(I,1).LT.1.E-5) GO TO 130
      IF (Y(I,1).LT.(-2.*Y(I,2))) GO TO 290
  130 ERROR(I)=0.
      M=0
      CALL DIFFUN (N,T,Y,W1(1,2),1,TOLD)
      NFE=NFE+1
      IF (IWEVAL.LE.0) GO TO 250
      IWEVAL=0
      RC=1.
      NJE=NJE+1
      NSTEPJ=NSTEP
      CON=-H*EL(1)
      ISV=M
      LSV=L
      NZ=IA(N+1)-1
      DO 140 I=1,NZ
  140 W2(I)=0.
      DO 220 IR=1,NR
      IF (KR(IR,1).EQ.0.OR.KR(IR,1).EQ.99) GO TO 220
      MT=ITYPE(IR)
      DO 150 I=1,MT
      JX=I+1-I/3*3
      KX=I+2-I/2*3
      J=KR(IR,JX)
      K=KR(IR,KX)
      IF (J.EQ.0) XX=1.
      IF (J.GT.0) XX=Y(J,1)
      IF (K.EQ.0) XK=1.
      IF (K.GT.0) XK=Y(K,1)
  150 RT(I)=R(IR)*XX*XK
      DO 210 K=1,MT
      I=KR(IR,K)
      DO 170 L=1,MT
      J=KR(IR,L)
      M=IA(J)-1
  160 M=M+1
      IF (I-JA(M)) 160,170,160
  170 W2(M)=W2(M)-RT(L)
      DO 200 L=4,12
      J=KR(IR,L)
      M=IA(I)-1
      IF (J) 200,210,180
  180 M=M+1
      IF (J-JA(M)) 180,190,180
  190 W2(M)=W2(M)+RT(K)*SC(IR,L)
  200 CONTINUE
  210 CONTINUE
  220 CONTINUE
      DO 240 J=1,N
      KMIN=IA(J)
      KMAX=IA(J+1)-1
      DO 230 K=KMIN,KMAX
      W2(K)=W2(K)*CON
      IF (JA(K).EQ.J) W2(K)=W2(K)+1.-CON*DILUT-CON*DNOWS
  230 CONTINUE
  240 CONTINUE
      CALL NSCORA (N,IA,JA.W2,IW1(1,2),W2(IPTR3),W2(IPTR2),IW1,IW1(1,7),
     1IW1(1,8))
      CALL NSNFAC (N,IW1(1,2),IW2,W2,IW1(1,3),IW2(IPTI2),IW1(1,4),W2(IPT
     1R2),W1(1,3),IW1(1,5),IW2(IPTI3),IW1(1,6),W2(IPTR3),W1,IW1(1,7),IW1
     2(1,8),IER)
      M=ISV
      L=LSV
      IF (IER.NE.0) GO TO 290
  250 DO 260 I=1,N
  260 W1(I,1)=H*W1(I,2)-(Y(I,2)+ERROR(I))
      CALL NSBSLV (N,IW1,IW1,IW1(1,3),IW2(IPTI2),IW1(1,4),W2(IPTR2),W1(1
     1,3),IW1(1,5),IW2(IPTI3),IW1(1,6),W2(IPTR3),W1(1,2),W1,W2)
      D=0.
      DO 270 I=1,N
      ERROR(I)=ERROR(I)+W1(I,2)
      D=D+(W1(I,2)/YMAX(I))**2
  270 W1(I,1)=Y(I,1)+EL(1)*ERROR(I)
      IF (M.NE.0) CRATE=AMAX1(.9*CRATE,D/D1)
      IF ((D*AMIN1(1.,2.*CRATE)).LE.BND) GO TO 320
      D1=D
      M=M+1
      IF (M.EQ.3) GO TO 280
      CALL DIFFUN (N,T,W1,W1(1,2),1,TOLD)
      GO TO 250
  280 NFE=NFE+2
      IF (IWEVAL.EQ.(-1)) GO TO 310
  290 T=TOLD
      RMAX=2.
      DO 300 J1=1,NQ
      DO 300 J2=J1,NQ
      J=(NQ+J1)-J2
      DO 300 I=1,N
  300 Y(I,J)=Y(I,J)-Y(I,J+1)
      IF (ABS(H).LE.HMIN*1.00001) GO TO 540
      RH=.25
      IREDO=1
      GO TO 70
  310 IWEVAL=MITER
      GO TO 120
  320 IF (MITER.NE.0) IWEVAL=-1
      NFE=NFE+M
      D=0.
      DO 330 I=1,N
  330 D=D+(ERROR(I)/YMAX(I))**2
      IF (D.GT.E) GO TO 360
      KFLAG=0
      IREDO=0
      NSTEP=NSTEP+1
      HUSED=H
      NQUSED=NQ
      DO 340 J=1,L
      DO 340 I=1,N
  340 Y(I,J)=Y(I,J)+EL(J)*ERROR(I)
      IF (IDOUB.EQ.1) GO TO 380
      IDOUB=IDOUB-1
      IF (IDOUB.GT.1) GO TO 560
      IF (L.EQ.LMAX) GO TO 560
      DO 350 I=1,N
  350 Y(I,LMAX)=ERROR(I)
      GO TO 560
  360 KFLAG=KFLAG-1
      T=TOLD
      DO 370 J1=1,NQ
      DO 370 J2=J1,NQ
      J=(NQ+J1)-J2
      DO 370 I=1,N
  370 Y(I,J)=Y(I,J)-Y(I,J+1)
      RMAX=2.
      IF (ABS(H).LE.HMIN*1.00001) GO TO 520
      IF (KFLAG.LE.(-3)) GO TO 500
      IREDO=2
      PR3=1.E+20
      GO TO 400
  380 PR3=1.E+20
      IF (L.EQ.LMAX) GO TO 400
      D1=0.
      DO 390 I=1,N
  390 D1=D1+((ERROR(I)-Y(I,LMAX))/YMAX(I))**2
      ENQ3=.5/FLOAT(L+1)
      PR3 = ((D1/EUP)**ENQ3) * 1.4 + 1.4E-6
  400 ENQ2=.5/FLOAT(L)
      PR2=((D/E)**ENQ2)*1.2+1.2E-6
      PR1=1.E+20
      IF (NQ.EQ.1) GO TO 420
      D=0.
      DO 410 I=1,N
  410 D=D+(Y(I,L)/YMAX(I))**2
      ENQ1=.5/FLOAT(NQ)
      PR1=((D/EDN)**ENQ1)*1.3+1.3E-6
  420 IF (PR2.LE.PR3) GO TO 430
      IF (PR3.LT.PR1) GO TO 450
      GO TO 440
  430 IF (PR2.GT.PR1) GO TO 440
      NEWQ=NQ
      RH=1./PR2
      GO TO 480
  440 NEWQ=NQ-1
      RH=1./PR1
      GO TO 480
  450 NEWQ=L
      RH=1./PR3
      IF (RH.LT.1.1) GO TO 470
      DO 460 I=1,N
  460 Y(I,NEWQ+1)=ERROR(I)*EL(L)/FLOAT(L)
      GO TO 490
  470 IDOUB=10
      GO TO 560
  480 IF ((KFLAG.EQ.0).AND.(RH.LT.1.1)) GO TO 470
      IF (NEWQ.EQ.NQ) GO TO 70
  490 NQ=NEWQ
      L=NQ+1
      IRET=2
      GO TO 30
  500 IF (KFLAG.EQ.(-9)) GO TO 530
      RH=10.**KFLAG
      RH=AMAX1(HMIN/ABS(H),RH)
      H=H*RH
      CALL DIFFUN (N,T,Y,W1,1,TOLD)
      NFE=NFE+1
      DO 510 I=1,N
  510 Y(I,2)=H*W1(I,1)
      IWEVAL=MITER
      IDOUB=10
      IF (NQ.EQ.1) GO TO 100
      NQ=1
      L=2
      IRET=3
      GO TO 30
  520 KFLAG=-1
      GO TO 560
  530 KFLAG=-2
      GO TO 560
  540 KFLAG=-3
      GO TO 560
  550 RMAX=100.
  560 HOLD=H
      JSTART=NQ
      RETURN
      END
      SUBROUTINE DIFFUN (L,T,X,XT,IENTRY,TXLD)
C
C  THIS ROUTINE CALCULATES THE RATE OF CHANGE OF EACH SPECIES DUE TO
C  DILUTION, EMISSIONS, SURFACE DEPOSITION, AND CHEMISTRY
C
      SAVE
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /HEAT/ SC(135,12),ISC(135,3)
      COMMON /NEED1/ IBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,IIO3,
     1IIH2O,IBZA,JPLUS
      COMMON /STORE/ AST(60)
      COMMON /TEMPER/ TEMEND,NTEMP,QM(30)
      COMMON /SUNLIT/ ZZ1(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),KK1(24),JSRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /PHOTON/ CF(72,20),P(24,20),IPH(20),IP,RFCT(20),PP(10,20),
     1IDEPO,RDEPO(26,10),LOCDEP(10),ISPDP(10),RDCOEF(125,10),IDPTIM,DPEN
     2D,RDEP1(26,10),DNOWS,SPRSE(300)
      COMMON /EMIS/ NEM,IS1,ESTRT(5),ESTOP,ESLP,EMSP(5),IEMLS(5),EOSLP(5
     1),EMO(26,5),ECI(5),EM(26),EC(125),ES(125,5)
      COMMON /MIX/ NMIX,AMIX(26),STRM,STOPM,DC(104)
      COMMON /MIXING/ DSTRT,DEND,AMC(5),BMC(5),CMC(5),FD(6),FG(6),
     1AMXX(26),DL,TTMAX,SRISE,SRMIN,DELH,TDIL,NMXX,HT,SSET,SSRISE
      COMMON /ALOFT/ IALFT,CALFT(10),LOCALF(10),ISPAL(10)
      COMMON /CRED/ ICR,ISPCR,ISPNCR(3),SPCR69(3),SURFCR(3),ALOFCR(3),
     1REDCR(3),FSRFCR(3),FALFCR(3),COSFBK,COAFBK
      DIMENSION XT(L), X(L), ENOWS(5)
      INTEGER SPECIS,EMSP,PSPEC,PLSP,HCSPEC
      COMMON /BK1/ FBK(10),FBKAL(5),HCBK,XNBK,OZBK,H2OBK
C
C     EMULATE MULTIPLE ENTRY WITH COMPUTED GO TO
      GO TO (10,380),IENTRY
C
C     MAIN ENTRY POINT
   10 CONTINUE
      N=L
      DILU=DILUT
      IF(T.LT.TSRT.OR.TXLD.GT.TSRT+DTIM) DILU=0.
      IF(T.LT.TSRT.OR.TXLD.GT.TSRT+DTIM) GO TO 60
      IF (AMIX(1).GT.(-1.)) GO TO 50
      TT=CLOCK(TCI,IFIX(T))
      TT=CLKMIN(TT)
      TT=TT-SRISE
      FDT=TT/DL
      IF(FDT.LT.0.0) FDT=0.0
      K=0
   20 K=K+1
      IF(FDT.GE.FD(K).AND.FDT.LT.FD(K+1)) GO TO 30
      IF(K.LT.5) GO TO 20
      IF(FDT.LT.FD(2)) GO TO 40
      FGG=1.0
      FGSLP=0.0
      GO TO 40
   30 CONTINUE
      XFD=FDT-FD(K)
      FGG=((AMC(K)*XFD+BMC(K))*XFD+CMC(K))*XFD+FG(K)
      FGSLP=(3.*AMC(K)*XFD+2.*BMC(K))*XFD+CMC(K)
   40 HT=SRMIN+FGG*DELH
      DILU=(FGSLP*DELH)/(HT*DL)
      GO TO 60
   50 IF (AMIX(1).LT.0.) GO TO 60
C
C           CALCULATE DILUTION FOR CURRENT TIME STEP
C
      I=IFIX(T/60.+1.9999995)
      IF (T.LE.TOLD.AND.I.LT.IOLD) I=IOLD
      IOLD=I
      IF (I.LE.1) I=2
      Z=T-FLOAT(I-2)*60.
      HT=((Z*DC(3*I-3)+DC(3*I-4))*Z+DC(3*I-5))*Z+AMIX(I-1)
      DILU=((3.0*Z*DC(3*I-3)+2.0*DC(3*I-4))*Z+DC(3*I-5))/HT
   60 DILUT=AMAX1(DILU,0.)
      TNOW=TEMP
      ZNOW=ZENI
      DO 70 I=1,N
      XT(I)=-DILUT*X(I)
      IF (SPECIS(I).EQ.IICO) XT(I)=XT(I)+DILUT*COAFBK
      IF (SPECIS(I) .EQ. IIH2O ) XT(I)=XT(I)+DILUT*X(I)
   70 IF (SPECIS(I).EQ.IIO3   ) XT(I)=DILUT*(OZAL-X(I))
      IF (T.LT.TSRT.OR.T.GT.TSRT+DTIM) GO TO 140
      DO 80 K=1,2
      J=INOX(K)
      IF (SPECIS(J).NE.IINO2 ) GO TO 80
      XT(J)=DILUT*(XNAL-X(J))
      GO TO 90
   80 CONTINUE
   90 DO 100 K=1,NHC
      J=IH(K)
      FBKS=FALHC(K)
      IF (FBK(K).GT.(-0.00001)) FBKS=FBK(K)
      XT(J)=DILUT*(FALHC(K)*HCAL/CARB(K)-X(J))
  100 CONTINUE
C
C  ENTRAIN OTHER SPECIES
```

```
C
      IF (IALFT.LE.0) GO TO 120
      DO 110 K=1,IALFT
      J=LOCALF(K)
      XT(J)=DILUT*(CALFT(K)-X(J))
  110 CONTINUE
  120 IF (ICR.LE.0) GO TO 140
      DO 130 JJ=1,ICR
      J=IEMLS(JJ+2)
      XT(J)=DILUT*(ALOFCR(JJ)-X(J))
  130 CONTINUE
C
C  INPUT EMISSIONS
C
  140 IF (T.GE.ESTOP) GO TO 230
      FNOW=Z1/HT
      IF (AMIX(1).GE.0.) FNOW=AMIX(1)/HT
  150 I=IFIX(T/60.)
      ENOW=0.
      ZZ=(T-FLOAT(I*60))/60.
      I=(I+1)*5
      IF (NEM.GT.0) ENOW=((((ZZ*EC(I)+EC(I-1))*ZZ+EC(I-2))*ZZ+EC(I-3
     1))*ZZ+EC(I-4))*FNOW/60.
      DO 160 J=1,ISP
      IF (NEM.GT.0.AND.J.LE.2) ENOWS(J)=ENOW
      IF (NEM.LE.-1)
     1ENOWS(J)=((((ZZ*ES(I,J)+ES(I-1,J))*ZZ+ES(I-2,J))*ZZ
     2  +ES(I-3,J))*ZZ+ES(I-4,J))*FNOW/60.
  160 CONTINUE
  170 DO 220 LL=1,ISP
      IF (EMSP(LL).NE.IINX ) GO TO 190
      DO 180 K=1,2
      J=INOX(K)
  180 XT(J)=XT(J)+ENOWS(LL)*EXN*FENX(K)
      GO TO 220
  190 IF (EMSP(LL).NE.IIHC  ) GO TO 210
      DO 200 K=1,NHC
      J=IH(K)
  200 XT(J)=XT(J)+ENOWS(LL)*EHC*RCTY(K)/CARB(K)
      GO TO 220
  210 J=IEMLS(LL)
      XT(J)=XT(J)+ENOWS(LL)*ECI(LL)
  220 CONTINUE
C
C  DO PHOTOLYSIS
C
  230 IF (T.EQ.TOLD) GO TO 290
      IF (IP.EQ.0) GO TO 280
      TC=CLOCK(TCI,IFIX(T))
      IF (TC.GT.2400.) TC=TC-2400.
      IF (TC.GE.SSRISE.AND.TC.LE.SSET) GO TO 250
      DO 240 JK=1,IP
      IR=IPH(JK)
      R(IR)=0.
  240 CONTINUE
      ZNOW=90.
      GO TO 280
  250 I=IFIX(T/60.+2.0)
      IF (I.GT.24) I=24
      Z=T/60.-FLOAT(I-2)
      IR=IPH(1)
      R1=(((Z*CF(3*I-3,1)+CF(3*I-4,1))*Z+CF(3*I-5,1))*Z+P(I-1,1))*A(IR)
      IF (R1.LT.0.0) R1=0.
      R(IR)=R1
      DO 270 J=2,IP
      IR=IPH(J)
      IF (PP(1,J).LT.0.) GO TO 260
      R(IR)=((Z*CF(3*I-3,J)+CF(3*I-4,J))*Z+CF(3*I-5,J))*Z+P(I-1,J)
      R(IR)=R(IR)*RFCT(J)*R1
      GO TO 270
  260 CONTINUE
      R(IR)=R1*RFCT(J)
  270 IF (R(IR).LT.0.) R(IR)=0.
      IZ=IFIX(T/60.)+1
      Z=T/60.-FLOAT(IZ-1)
      IF (T.LT.60.) ZNOW=XZ(1)+(0.5*XZ(3)*(Z-1.)+0.5*XZ(1)*(Z-3.)
     1  -XZ(2)*(Z-2.))*Z
      IF (T.GE.60..AND.IZ.LT.23)
     1 ZNOW=XZ(IZ)+0.25*Z*((5.*XZ(IZ+1)-3.*XZ(IZ)-XZ(IZ-1)
     2  -XZ(IZ+2))+(XZ(IZ-1)-XZ(IZ)-XZ(IZ+1)+XZ(IZ+2))*Z)
      IF (IZ.GT.22) ZNOW=(XZ(IZ+1)-XZ(IZ))*Z+XZ(IZ)
C
C  DO TEMPERATURES
C
  280 IF (T.GT.TEMEND) GO TO 290
      IZ=IFIX(T*TINV)+1
      Z=T*TINV-FLOAT(IZ-1)
      IF (T.LE.60.) TNOW=QM(1)+(0.5*QM(3)*(Z-1.)+0.5*QM(1)*(Z-3.)
     1  -QM(2)*(Z-2.))*Z
      IF (T.GT.60.) TNOW=QM(IZ)+0.25*Z*((5.*QM(IZ+1)-3.*QM(IZ)-QM(IZ-1)
     1  -QM(IZ+2))+(QM(IZ-1)-QM(IZ)-QM(IZ+1)+QM(IZ+2))*Z)
      IF (TNOW.NE.TEMP.AND.TNOW.GT.0.) FCT=((3355.7046E-6)*TNOW-1.)/TNO
     1W
C
C  DO SURFACE DEPOSITION
C
  290 IF (T.GE.DPEND) GO TO 330
      FNOW=1./HT
C     IF (AMIX(1).GE.0.) FNOW=AMIX(1)/HT
      I=IFIX(T/60.)
      Z=(T-FLOAT(I*60))/60.
      I=(I+1)*5
      DO 320 J=1,IDEPO
      DNOWS=((((Z*RDCOEF(I,J)+RDCOEF(I-1,J))*Z+RDCOEF(I-2,J))*Z
     1  +RDCOEF(I-3,J))*Z+RDCOEF(I-4,J))*FNOW/60.
      IF (ISPDP(J).NE.IIHC) GO TO 310
      DO 300 K=1,NHC
      LL=IH(K)
  300 XT(LL)=XT(LL)-DNOWS*X(LL)*RCTY(K)/CARB(K)
      GO TO 320
  310 LL=LOCDEP(J)
      XT(LL)=XT(LL)-DNOWS*X(LL)
  320 CONTINUE
C
C  DO REACTION RATES
C
  330 DO 370 IR=1,NR
      I=KR(IR,1)
      IF (I.EQ.0.OR.R(IR).EQ.0.) GO TO 370
      RT=1.
      JT=ITYPE(IR)
      DO 340 LL=1,JT
      I=KR(IR,LL)
      IF (I.NE.99.AND.I.NE.0) RT=RT*X(I)
  340 CONTINUE
      IF (ABS(S(IR)).NE.0.) R(IR)=A(IR)*EXP(S(IR)*FCT)
      RT=RT*R(IR)
      DO 350 LL=1,JT
      I=KR(IR,LL)
      IF (I.NE.0) XT(I)=XT(I)-RT
  350 CONTINUE
      DO 360 K=4,12
      I=KR(IR,K)
      IF (I.EQ.0) GO TO 370
      XT(I)=XT(I)+SC(IR,K)*RT
  360 CONTINUE
  370 CONTINUE
      TOLD=T
      TEMP=TNOW
      ZENI=ZNOW
      RETURN
C     ENTRY DIFSET
  380 CONTINUE
      TOLD=T
      TINV=1./60.
      FCT=((3355.7046E-6)*TEMP-1.)/TEMP
      IOLD=0
      TCI=FLOAT(JSRT)
      DHC=HCAL-HCBK
      DXN=XNAL-XNBK
      DO3=OZAL-OZBK
      IF (DHC.LT.0.) HCAL=HCBK
      IF (DXN.LT.0.) DXN=0.
      IF (DO3.LT.0.) DO3=0.
      HT=AMIX(1)
      IF (HT.LT.0.) HT=AMXX(1)
      IF (IP.EQ.0) RETURN
      II=0
      DO 390 J=1,IP
      IR=IPH(J)
      R(IR)=P(1,J)*RFCT(J)
      IF (J.GT.1) R(IR)=R(IR)*R(1)
      IF (R(IR).LT.0.) R(IR)=0.
  390 CONTINUE
      ZENI=XZ(1)
      ISP=IS1+ICR
      RETURN
      END
      SUBROUTINE COSET (METH,NQ,EL,TQ,MAXDER)
C
C  PART OF THE GEAR ROUTINES
C
      SAVE
      DIMENSION PERTST(12,2,3), EL(13), TQ(4)
      DATA PERTST/1.,1.,2.,1.,.3158,.07407,.01391,.002182,.0002945,.0000
     13492,.000003692,.0000003524,1.,1.,.5,.1667,.04167,1.,1.,1.,1.,1.,1
     2.,1.,2.,12.,24.,37.89,53.33,70.08,87.97,106.9,126.7,147.4,168.8,19
     31.0,2.0,4.5,7.333,10.42,13.7,1.,1.,1.,1.,1.,1.,1.,12.0,24.0,37.89,
     453.33,70.08,87.97,106.9,126.7,147.4,168.8,191.0,1.,3.0,6.0,9.167,1
     52.5,1.,1.,1.,1.,1.,1.,1.,1./
   10 MAXDER=5
      GO TO (20,30,40,50,60),NQ
   20 EL(1)=1.0
      GO TO 70
   30 EL(1)=6.6666666666667E-01
      EL(3)=3.3333333333333E-01
      GO TO 70
   40 EL(1)=5.4545454545455E-01
      EL(3)=EL(1)
      EL(4)=9.0909090909091E-02
      GO TO 70
   50 EL(1)=0.48
      EL(3)=0.7
      EL(4)=0.2
      EL(5)=0.02
      GO TO 70
   60 EL(1)=4.3795620437956E-01
      EL(3)=8.2116788321168E-01
      EL(4)=3.1021897810219E-01
      EL(5)=5.4744525547445E-02
      EL(6)=3.6496350364964E-03
   70 DO 80 K=1,3
   80 TQ(K)=PERTST(NQ,METH,K)
      TQ(4)=.5*TQ(2)/FLOAT(NQ+2)
      RETURN
      END
      FUNCTION CLKMIN (X)
C
C  CALCULATE THE NUMBER OF MINUTES BASED ON A 24 HR CLOCK
C
      SAVE
      JSTRT=IFIX(X+0.1)
      JMIN=JSTRT-((JSTRT/100)*100)
      CLKMIN=FLOAT((JSTRT/100)*60+JMIN)
      RETURN
      END
      SUBROUTINE INTERP (TOUT,Y,NO,YO)
C
C  PERFORM EXACT INTERPOLATION IN TIME
C
      SAVE
      COMMON /GEAR1/ T,H,DUMMY(4),N,IDUMMY(2),JSTART
      DIMENSION YO(NO), Y(NO,1)
      DO 10 I=1,N
   10 YO(I)=Y(I,1)
      L=MAXO(JSTART+1,2)
      S=(TOUT-T)/H
      S1=1.
      DO 30 J=2,L
      S1=S1*S
      DO 20 I=1,N
   20 YO(I)=YO(I)+S1*Y(I,J)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE NSBSLV (N,R,C,IL,JL,ISL,L,D,IU,JU,ISU,U,X,B,Y)
C
C  PART OF THE GEAR ROUTINES
C
      SAVE
      DIMENSION R(1), IL(1), JL(1), IU(1), JU(1), C(1), ISL(1), ISU(1)
      DIMENSION L(1), X(1), B(1), U(1), Y(1), D(1)
      INTEGER R,RK,C,CK
      REAL L
      DO 10 K=1,N
      RK=R(K)
   10 Y(K)=B(RK)
      DO 30 K=1,N
      JMIN=IL(K)
      JMAX=IL(K+1)-1
      YK=-D(K)*Y(K)
      Y(K)=-YK
      IF (JMIN.GT.JMAX) GO TO 30
      ISLB=ISL(K)-1
      DO 20 J=JMIN,JMAX
      ISLB=ISLB+1
      JLJ=JL(ISLB)
   20 Y(JLJ)=Y(JLJ)+YK*L(J)
   30 CONTINUE
      K=N
      DO 60 I=1,N
      SUM=-Y(K)
      JMIN=IU(K)
      JMAX=IU(K+1)-1
      IF (JMIN.GT.JMAX) GO TO 50
      ISUB=ISU(K)-1
      DO 40 J=JMIN,JMAX
      ISUB=ISUB+1
      JUJ=JU(ISUB)
   40 SUM=SUM+U(J)*Y(JUJ)
   50 Y(K)=-SUM
      CK=C(K)
      X(CK)=-SUM
   60 K=K-1
      RETURN
      END
      SUBROUTINE NSCORA (N,IA,JA,A,IAP,JAWORK,AWORK,C,IR,ICT)
C
C  PART OF THE GEAR ROUTINES
C
      SAVE
      INTEGER IA(1),JA(1),IAP(1),JAWORK(1),C(1),IR(1),ICT(1)
      REAL A(1),AWORK(1)
      DO 10 K=1,N
      ICK=C(K)
   10 IR(ICK)=K
      JMIN=1
      DO 30 K=1,N
      ICK=C(K)
      JMAX=JMIN+IA(ICK+1)-IA(ICK)-1
      IF (JMIN.GT.JMAX) GO TO 30
      IAINK=IA(ICK)-1
      DO 20 J=JMIN,JMAX
      IAINK=IAINK+1
      JAOUTJ=JA(IAINK)
      JAOUTJ=IR(JAOUTJ)
      JAWORK(J)=JAOUTJ
   20 AWORK(J)=A(IAINK)
   30 JMIN=JMAX+1
      DO 40 I=1,N
   40 ICT(I)=IAP(I)
      JMIN=1
      DO 60 I=1,N
      ICK=C(I)
      JMAX=JMIN+IA(ICK+1)-IA(ICK)-1
      IF (JMIN.GT.JMAX) GO TO 60
      DO 50 J=JMIN,JMAX
      JAOUTJ=JAWORK(J)
      ICTJ=ICT(JAOUTJ)
      A(ICTJ)=AWORK(J)
      ICT(JAOUTJ)=ICTJ+1
   50 CONTINUE
   60 JMIN=JMAX+1
      RETURN
      END
      SUBROUTINE NSCORD (N,IA,JA,IAWORK,JAWORK,C,IR,ICT)
C
C   PART OF THE GEAR ROUTINES
C
      SAVE
      INTEGER IA(1),JA(1),IAWORK(1),JAWORK(1),C(1),IR(1),ICT(1)
      DO 10 I=1,N
   10 ICT(I)=0
      IAWORK(1)=1
      DO 30 K=1,N
      ICK=C(K)
      JMIN=IAWORK(K)
      JMAX=JMIN+IA(ICK+1)-IA(ICK)-1
      IAWORK(K+1)=JMAX+1
      IF (JMIN.GT.JMAX) GO TO 30
      IAINK=IA(ICK)-1
      DO 20 J=JMIN,JMAX
      IAINK=IAINK+1
      JAOUTJ=JA(IAINK)
      JAOUTJ=IR(JAOUTJ)
      JAWORK(J)=JAOUTJ
   20 ICT(JAOUTJ)=ICT(JAOUTJ)+1
   30 CONTINUE
      IA(1)=1
      DO 40 I=1,N
      IA(I+1)=IA(I)+ICT(I)
   40 ICT(I)=IA(I)
      DO 60 I=1,N
      JMIN=IAWORK(I)
      JMAX=IAWORK(I+1)-1
      IF (JMIN.GT.JMAX) GO TO 60
      DO 50 J=JMIN,JMAX
      JAOUTJ=JAWORK(J)
      ICTJ=ICT(JAOUTJ)
      JA(ICTJ)=I
   50 ICT(JAOUTJ)=ICTJ+1
   60 CONTINUE
      RETURN
      END
      SUBROUTINE NSNFAC (N,IA,JA,A,IL,JL,ISL,L,D,IU,JU,ISU,U,X,IRL,JRL,I
     1ER)
C
C  PART OF THE GEAR ROUTINES
C
      SAVE
      INTEGER IA(1),JA(1),IL(1),JL(1),ISL(1)
      INTEGER IU(1),JU(1),ISU(1),IRL(1),JRL(1)
      REAL A(1),L(1),D(1),U(1),X(1)
      REAL LKI
      IER=0
      DO 10 K=1,N
      IRL(K)=IL(K)
   10 JRL(K)=0
      DO 180 K=1,N
      X(K)=0.
      I1=0
      IF (JRL(K).EQ.0) GO TO 30
      I=JRL(K)
   20 I2=JRL(I)
      JRL(I)=I1
      I1=I
      X(I)=0.
      I=I2
      IF (I.NE.0) GO TO 20
   30 JMIN=ISU(K)
      JMAX=JMIN+IU(K+1)-IU(K)-1
      IF (JMIN.GT.JMAX) GO TO 50
      DO 40 J=JMIN,JMAX
      JUJ=JU(J)
   40 X(JUJ)=0.
   50 JMIN=IA(K)
      JMAX=IA(K+1)-1
      DO 60 J=JMIN,JMAX
      JAJ=JA(J)
   60 X(JAJ)=A(J)
      I=I1
      IF (I.EQ.0) GO TO 100
   70 IRLI=IRL(I)
      LKI=-X(I)
      L(IRLI)=-LKI
      JMIN=IU(I)
      JMAX=IU(I+1)-1
      IF (JMIN.GT.JMAX) GO TO 90
      ISUB=ISU(I)-1
      DO 80 J=JMIN,JMAX
      ISUB=ISUB+1
      JUJ=JU(ISUB)
   80 X(JUJ)=X(JUJ)+LKI*U(J)
   90 I=JRL(I)
```

```
    IF (I.NE.0) GO TO 70
100 IF (X(K).EQ.0.) GO TO 190
    DK=1./X(K)
    D(K)=DK
    IF (K.EQ.N) GO TO 180
    JMIN=IU(K)
    JMAX=IU(K+1)-1
    IF (JMIN.GT.JMAX) GO TO 120
    ISUB=ISU(K)-1
    DO 110 J=JMIN,JMAX
    ISUB=ISUB+1
    JUJ=JU(ISUB)
110 U(J)=X(JUJ)*DK
120 CONTINUE
    I=I1
    IF (I.EQ.0) GO TO 170
130 IRL(I)=IRL(I)+1
    I1=JRL(I)
    IF (IRL(I).GE.IL(I+1)) GO TO 160
    ISLB=IRL(I)-IL(I)+ISL(I)
    J=JL(ISLB)
140 IF (I.GT.JRL(J)) GO TO 150
    J=JRL(J)
    GO TO 140
150 JRL(I)=JRL(J)
    JRL(J)=I
160 I=I1
    IF (I.NE.0) GO TO 130
170 ISLK=ISL(K)
    IF (IRL(K).GE.IL(K+1)) GO TO 180
    J=JL(ISLK)
    JRL(K)=JRL(J)
    JRL(J)=K
180 CONTINUE
    RETURN
190 IER=K
    RETURN
    END
```

```
      SUBROUTINE NSSFAC (N,IA,JA,MAXPL,IL,JL,ISL,MAXPU,IU,JU,ISU,P,V,IRA
     1,JRA,IRAC,IRL,JRL,IRU,JRU,IER)
C
C  PART OF THE GEAR ROUTINES
C
      SAVE
      INTEGER IA(1),JA(1),IL(1),JL(1),ISL(1)
      INTEGER IU(1),JU(1),ISU(1),IRL(1),JRL(1),V(1)
C
C  NOTE THE FOLLOWING STATEMENT DECLARATION MUST BE CHANGED TO "REAL"
C   INSTEAD OF "INTEGER" ON  IBM, AMDAHL, AND HP COMPUTERS
C
      INTEGER P(1),IRAC(1),IRA(1),JRA(1),IRU(1),JRU(1)
      INTEGER VI,VJ,PK,PPK,PI,CEND,REND
      IER=0
      DO 10 K=1,N
      IRAC(K)=0
      JRA(K)=0
      JRL(K)=0
   10 JRU(K)=0
      DO 20 K=1,N
      IAK=IA(K)
      IF (IAK.GT.IA(K+1)) GO TO 400
      IF (JA(IAK).GT.K) GO TO 20
      JAIAK=JA(IAK)
      JRA(K)=IRAC(JAIAK)
      IRAC(JAIAK)=K
   20 IRA(K)=IAK
      JLPTR=0
      IL(1)=1
      JUPTR=0
      IU(1)=1
      DO 390 K=1,N
      P(1)=1
      V(1)=N+1
      LSFS=2
      VJ=IRAC(K)
      IF (VJ.EQ.0) GO TO 60
   30 PPK=1
   40 PK=PPK
      PPK=P(PK)
      IF (V(PPK)-VJ) 40,440,50
   50 P(PK)=LSFS
      V(LSFS)=VJ
      P(LSFS)=PPK
      LSFS=LSFS+1
      VJ=JRA(VJ)
      IF (VJ.NE.0) GO TO 30
   60 LASTI=0
      I=K
   70 I=JRU(I)
      IF (I.EQ.0) GO TO 120
      PPK=1
      JMIN=IRL(I)
      JMAX=ISL(I)+IL(I+1)-IL(I)-1
      IF (LASTI.GT.I) GO TO 80
      LASTI=I
      LASTID=JMAX-JMIN
      IF (JL(JMIN).NE.K) LASTID=LASTID+1
   80 IF (JMIN.GT.JMAX) GO TO 70
      DO 110 J=JMIN,JMAX
      VJ=JL(J)
   90 PK=PPK
      PPK=P(PK)
      IF (V(PPK)-VJ) 90,110,100
  100 P(PK)=LSFS
      V(LSFS)=VJ
      P(LSFS)=PPK
      PPK=LSFS
      LSFS=LSFS+1
  110 CONTINUE
      GO TO 70
  120 PI=P(1)
      IF (V(PI).NE.K) GO TO 450
      IF (LASTI.EQ.0) GO TO 130
      IF (LASTID.NE.LSFS-3) GO TO 130
      IRLL=IRL(LASTI)
      ISL(K)=IRLL+1
      IF (JL(IRLL).NE.K) ISL(K)=ISL(K)-1
      IL(K+1)=IL(K)+LASTID
      IRL(K)=ISL(K)
      GO TO 160
  130 ISL(K)=JLPTR+1
      PI=P(1)
      PI=P(PI)
      VI=V(PI)
  140 IF (VI.GT.N) GO TO 150
      JLPTR=JLPTR+1
      IF (JLPTR.GT.MAXPL) GO TO 460
      JL(JLPTR)=VI
      PI=P(PI)
      VI=V(PI)
      GO TO 140
  150 IRL(K)=ISL(K)
      IL(K+1)=IL(K)+JLPTR-ISL(K)+1
  160 P(1)=1
      V(1)=N+1
      LSFS=2
      JMIN=IRA(K)
      JMAX=IA(K+1)-1
      IF (JMIN.GT.JMAX) GO TO 200
      DO 190 J=JMIN,JMAX
      VJ=JA(J)
      PPK=1
  170 PK=PPK
      PPK=P(PK)
      IF (V(PPK)-VJ) 170,410,180
  180 P(PK)=LSFS
      V(LSFS)=VJ
      P(LSFS)=PPK
  190 LSFS=LSFS+1
  200 LASTI=0
      I=K
  210 I=JRL(I)
      IF (I.EQ.0) GO TO 260
      PPK=1
      JMIN=IRU(I)
      JMAX=ISU(I)+IU(I+1)-IU(I)-1
      IF (LASTI.GT.I) GO TO 220
      LASTI=I
      LASTID=JMAX-JMIN
      IF (JU(JMIN).NE.K) LASTID=LASTID+1
  220 IF (JMIN.GT.JMAX) GO TO 210
      DO 250 J=JMIN,JMAX
      VJ=JU(J)
  230 PK=PPK
      PPK=P(PK)
      IF (V(PPK)-VJ) 230,250,240
  240 P(PK)=LSFS
      V(LSFS)=VJ
      P(LSFS)=PPK
      PPK=LSFS
  250 LSFS=LSFS+1
      GO TO 210
  260 PI=P(1)
      IF (V(PI).NE.K) GO TO 420
      IF (LASTI.EQ.0) GO TO 270
      IF (LASTID.NE.LSFS-3) GO TO 270
      IRUL=IRU(LASTI)
      ISU(K)=IRUL+1
      IF (JU(IRUL).NE.K) ISU(K)=ISU(K)-1
      IU(K+1)=IU(K)+LASTID
      IRU(K)=ISU(K)
      GO TO 300
  270 ISU(K)=JUPTR+1
      PI=P(1)
      PI=P(PI)
      VI=V(PI)
  280 IF (VI.GT.N) GO TO 290
      JUPTR=JUPTR+1
      IF (JUPTR.GT.MAXPU) GO TO 430
      JU(JUPTR)=VI
      PI=P(PI)
      VI=V(PI)
      GO TO 280
  290 IRU(K)=ISU(K)
      IU(K+1)=IU(K)+JUPTR-ISU(K)+1
  300 I=K
  310 I1=JRL(I)
      CEND=ISL(I)+IL(I+1)-IL(I)
      IF (IRL(I).GE.CEND) GO TO 320
      IRLI=IRL(I)
      J=JL(IRLI)
      JRL(I)=JRL(J)
      JRL(J)=I
  320 I=I1
      IF (I.EQ.0) GO TO 330
      IRL(I)=IRL(I)+1
      GO TO 310
  330 I=K
  340 I1=JRU(I)
      REND=ISU(I)+IU(I+1)-IU(I)
      IF (IRU(I).GE.REND) GO TO 350
      IRUI=IRU(I)
      J=JU(IRUI)
      JRU(I)=JRU(J)
      JRU(J)=I
  350 I=I1
      IF (I.EQ.0) GO TO 360
      IRU(I)=IRU(I)+1
      GO TO 340
  360 I=IRAC(K)
      IF (I.EQ.0) GO TO 390
  370 I1=JRA(I)
      IRA(I)=IRA(I)+1
      IF (IRA(I).GE.IA(I+1)) GO TO 380
      IRAI=IRA(I)
      IF (JA(IRAI).GT.I) GO TO 380
      JAIRAI=JA(IRAI)
      JRA(I)=IRAC(JAIRAI)
      IRAC(JAIRAI)=I
  380 I=I1
      IF (I.NE.0) GO TO 370
  390 CONTINUE
      ISL(N)=JLPTR
      ISU(N)=JUPTR
      RETURN
  400 CALL YSMER ('ROW',K,' OF A IS NULL')
      GO TO 470
  410 CALL YSMER ('ROW',K,' HAS DUPLICATE ENTRY')
```

```
    GO TO 470
420 CALL YSMER ('ROW',K,' HAS A NULL PIVOT')
    GO TO 470
430 CALL YSMER ('ROW',K,' EXCEEDS JU STORAGE')
    GO TO 470
440 CALL YSMER ('COL',K,' HAS DUPLICATE ENTRY')
    GO TO 470
450 CALL YSMER ('COL',K,' HAS A NULL PIVOT')
    GO TO 470
460 CALL YSMER ('COL',K,' EXCEEDS JL STORAGE')
470 IER=1
    RETURN
    END
```

```
      SUBROUTINE OZMX (C,TLL,TOUT,NO)
C
C  CALCULATE THE MAXIMUM 1 HR AVERAGE VALUE USING A RUNNING AVERAGE
C
      SAVE
      DIMENSION C(NO,6), RUN(61,5), OZ(61,5), OZA(5), TL(5), NT(5)
      DIMENSION S2(5),SIX(5)
      COMMON /GEAR1/ T,H,HMIN,HMAX,EPSC,UROUND,NC,MFC,KFLAG,JSTART
      COMMON /HOUR/ OZM(5),NGO,TTM,TM(5)
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),CR(61),REACT(61),NI,PLSP(5),JOZ(5)
      INTEGER SPECIS,HCSPEC,REACT,PLSP
      NQ=JSTART
      IF (T.NE.START) GO TO 20
      DO 10 I=1,NOZ
      KOZ=JOZ(I)
      RUN(1,I)=C(KOZ,1)
      NT(I)=1
      TL(I)=START+1.
      OZM(I)=0.
      OZA(I)=0.
   10 CONTINUE
      TLL=TL(1)
      RETURN
   20 DO 130 JJ=1,NOZ
      KOZ=JOZ(JJ)
   30 DO 40 I=1,61
      RR=(TL(JJ)+FLOAT(I-1)-T)/H
      IF (RR.GT.0.) GO TO 50
      IF (TL(JJ)+FLOAT(I-1).GT.STOPP) GO TO 50
      OZ(I,JJ)=C(KOZ,1)
      RH=1.
      DO 40 J=1,NQ
      RH=RH*RR
   40 OZ(I,JJ)=OZ(I,JJ)+RH*C(KOZ,J+1)
      I=61
   50 I=I-1
      TL(JJ)=TL(JJ)+FLOAT(I)
      DO 90 J=1,I
      IF (NT(JJ).GT.60) GO TO 70
      NTT=NT(JJ)
      RUN(NTT+1,JJ)=OZ(J,JJ)
      NT(JJ)=NT(JJ)+1
      IF (NT(JJ).LT.61) GO TO 90
      SIX(JJ)=0.
      S2(JJ)=RUN(60,JJ)
      DO 60 K=1,29
      SIX(JJ)=SIX(JJ)+RUN(2*K+1,JJ)
   60 S2(JJ)=S2(JJ)+RUN(2*K,JJ)
   70 OZA(JJ)=2.*SIX(JJ)+4.*S2(JJ)+RUN(1,JJ)+OZ(J,JJ)
      SV=SIX(JJ)
      SIX(JJ)=S2(JJ)-RUN(2,JJ)
      S2(JJ)=SV+OZ(J,JJ)
      RUN(61,JJ)=OZ(J,JJ)
      DO 80 K=1,60
   80 RUN(K,JJ)=RUN(K+1,JJ)
      IF (OZA(JJ).GT.OZM(JJ)) OZM(JJ)=OZA(JJ)
      IF (OZA(JJ).EQ.OZM(JJ)) TM(JJ)=TL(JJ)+FLOAT(J-I-31)
   90 CONTINUE
      IF (TL(JJ)-T.LT.0..AND.T.LT.STOPP) GO TO 30
      IF (NGO.NE.0) GO TO 100
      IF (T.GT.360..AND.OZA(JJ).LT.OZM(JJ)) GO TO 110
  100 IF (T.LT.STOPP) GO TO 130
      GO TO 120
  110 TOUT=T
      STOPP=T
  120 OZM(JJ)=OZM(JJ)/180.
      OZA(JJ)=OZA(JJ)/180.
      IF (OZA(JJ).LT.OZM(JJ)) GO TO 130
      IF (T.GT.STOPP.AND.OZM(JJ).GT.0.00001) TM(JJ)=-TM(JJ)
  130 CONTINUE
      TLL=TL(1)
      RETURN
      END
      SUBROUTINE SAVIT (T,C)
C
C  SAVE INTERMEDIATE CONCENTRATIONS FOR PLOTTING PURPOSES
C
      SAVE
      COMMON /FRPLOT/ SAVCON(80,5),SAVTIM(80),NT,INOW
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),CR(61),REACT(61),NI,PLSP(5),KOZ(5)
C
      DIMENSION C(80)
      DATA NFRST/1/
C
      IF (NFRST.NE.1.AND.T.EQ.TOLD) RETURN
      IF (NT.EQ.0) NT=1
      NFRST=2
      TOLD=T
      IF (NT.GT.80) RETURN
      IF (T.LT.SAVTIM(NT)) RETURN
      SAVTIM(NT)=T
C
      DO 10 II=1,NOZ
      I=KOZ(II)
      SAVCON(NT,II)=C(I)
   10 CONTINUE
      NT=NT+1
      RETURN
      END
      SUBROUTINE SORDER (N,IA,JA,P,Q,MAX,V,L,IER)
C
C  PART OF THE GEAR ROUTINES
C
      SAVE
      INTEGER IA(1),JA(1),P(1),Q(1),V(1),L(1)
      INTEGER S,SFS,PI,PJ,VI,VJ,VK,QVK,DTHR,DMIN
      IER=0
      DO 10 S=1,MAX
   10 L(S)=S+1
      SFS=1
      L(MAX)=0
      DO 20 K=1,N
      P(K)=K
      Q(K)=K
      V(K)=1
   20 L(K)=0
      SFS=SFS+N
      DO 100 K=1,N
      JMIN=IA(K)
      JMAX=IA(K+1)-1
      IF(JMIN.GT.JMAX+1) GO TO 290
      KDIAG=0
      DO 90 J=JMIN,JMAX
      VJ=JA(J)
      IF (VJ.NE.K) GO TO 30
      KDIAG=1
      GO TO 90
   30 LLK=K
   40 LK=LLK
      LLK=L(LK)
      IF (LLK.EQ.0) GO TO 50
      IF (V(LLK)-VJ) 40,60,50
   50 LLK=SFS
      IF (LLK.EQ.0) GO TO 300
      SFS=L(SFS)
      V(K)=V(K)+1
      V(LLK)=VJ
      L(LLK)=L(LK)
      L(LK)=LLK
   60 LLK=VJ
   70 LK=LLK
      LLK=L(LK)
      IF (LLK.EQ.0) GO TO 80
      IF (V(LLK)-K) 70,90,80
   80 LLK=SFS
      IF (LLK.EQ.0) GO TO 300
      SFS=L(SFS)
      V(VJ)=V(VJ)+1
      V(LLK)=K
      L(LLK)=L(LK)
      L(LK)=LLK
   90 CONTINUE
      IF (KDIAG.EQ.0) GO TO 320
  100 CONTINUE
      J=0
      DTHR=0
      DMIN=N
      I=0
  110 I=I+1
      IF (I.GT.N) GO TO 280
      JMIN=MAX0(J+1,I)
      IF (JMIN.GT.N) GO TO 140
  120 DO 130 J=JMIN,N
      VI=P(J)
      IF (V(VI).LE.DTHR) GO TO 150
      IF (V(VI).LT.DMIN) DMIN=V(VI)
  130 CONTINUE
  140 DTHR=DMIN
      DMIN=N
      JMIN=I
      GO TO 120
  150 PJ=P(I)
      P(J)=PJ
      Q(PJ)=J
      PI=VI
      P(I)=PI
      Q(PI)=I
      LI=VI
  160 LI=L(LI)
      IF (LI.EQ.0) GO TO 210
      VK=V(LI)
      LLK=VK
      LJ=VI
  170 LJ=L(LJ)
      IF (LJ.EQ.0) GO TO 200
      VJ=V(LJ)
      IF (VJ.EQ.VK) GO TO 170
  180 LK=LLK
      LLK=L(LK)
      IF (LLK.EQ.0) GO TO 190
      IF (V(LLK)-VJ) 180,170,190
  190 LLK=SFS
      IF (LLK.EQ.0) GO TO 310
      SFS=L(SFS)
      V(VK)=V(VK)+1
      V(LLK)=VJ
      L(LLK)=L(LK)
      L(LK)=LLK
      GO TO 170
```

```
200 IF (V(VK).GT.V(VI)) GO TO 160
    I=I+1
    QVK=Q(VK)
    PI=P(I)
    P(QVK)=PI
    Q(PI)=QVK
    P(I)=VK
    Q(VK)=I
    GO TO 160
210 LI=VI
220 IF (L(LI).EQ.0) GO TO 270
    LI=L(LI)
    VK=V(LI)
    LLK=VK
    QVK=MIN0(Q(VK),I)
230 LK=LLK
    LLK=L(LK)
    IF (LLK.EQ.0) GO TO 240
    VJ=V(LLK)
    IF (Q(VJ).GT.QVK) GO TO 230
    V(VK)=V(VK)-1
    L(LK)=L(LLK)
    L(LLK)=SFS
    SFS=LLK
    LLK=LK
    GO TO 230
240 IF (Q(VK).LE.I) GO TO 260
    IF (V(VK).LE.DTHR) GO TO 250
    IF ((DTHR.LT.V(VK)).AND.(V(VK).LT.DMIN)) DMIN=V(VK)
    GO TO 220
250 J=MIN0(Q(VK)-1,J)
    DTHR=V(VK)
    GO TO 220
260 L(LK)=SFS
    SFS=L(VK)
    GO TO 220
270 L(LI)=SFS
    SFS=L(VI)
    GO TO 110
280 RETURN
290 CALL YSMER ('ROW',K,' OF A IS NULL')
    GO TO 330
300 CALL YSMER ('ROW',K,' EXCEEDS STORAGE')
    GO TO 330
310 CALL YSMER ('VERTEX',VI,' EXCEEDS STORAGE')
    GO TO 330
320 CALL YSMER ('COLUMN',K,'.. DIAGONAL MISSING')
330 IER=1
    RETURN
    END
```

```
      SUBROUTINE YSMER (A,K,A1)
C
C  WRITE ERROR MESSAGES
C
      SAVE
      INTEGER A,A1(5)
      COMMON /INOUT/ INP,LOUT,ITAPE,IALN,IALL,INHH,IOZC
      WRITE (LOUT,10) A,K,(A1(II),II=1,5)
      RETURN
C
C
   10 FORMAT (1X,A4,I6,5A4)
      END
      FUNCTION CIRC (TI,XI,YI)
C
C  PERFORM INTERPOLATION USING FOUR POINTS
C
      SAVE
      DIMENSION XI(3), YI(3)
      DX=XI(3)-XI(1)
      DY=YI(3)-YI(1)
      X=(XI(2)-XI(1))/DX
      Y=(YI(2)-YI(1))/DY
      T=(TI-XI(1))/DX
      IF (X.EQ.Y) GO TO 20
      B=((X*X+Y*Y)*0.5-X)/(Y-X)
      TT=SQRT(B*B+2.*T*(1.-B)-T*T)
      CX=B+TT
      IF (X.GT.Y) CX=B-TT
      IF (CX.LT.0..OR.CX.GT.1.) GO TO 20
   10 CIRC=CX*DY+YI(1)
      RETURN
   20 CX=T*Y/X
      IF (T.GT.X) CX=(T-X)*(1.-Y)/(1.-X)+Y
      GO TO 10
      END
      SUBROUTINE SPLOT
C
C  PLOTS SPECIES CONCENTRATIONS AS A FUNCTION OF TIME
C
      SAVE
      COMMON /TITL/ ITTL(36)
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /SUNLIT/ Z(10),RTCON(10),
     1LAM1,INC,SLA,SLO,TZ,IY,IM,ID,ISTRT,ISTOP,IINC,IEND,SPECIE,MAXZ,ITI
     2ME(24),XZ(24),KKK(24),JSTRT,JSTOP,PSPEC,MNLM,MXLM,MAXL,MAXJ
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /HOUR/ OZM(5),NGO,TM,DUMMM(5)
      COMMON /FRPLOT/ SAVCON(80,5),SAVTIM(80),NT1,INOW
      COMMON /PLTND/ JBAR,JSYMB,CVERT(9),TVERT(52,2)
      COMMON /NEED1/ JBLANK,MBLANK,IIHC,IINX,IICO,IINO2,IINO,IIO3,
     1IIH2O,IBZA,JPLUS
      COMMON /GEAR7/ JGRID(101),JSAV(101),IDD(1798)
      DIMENSION TPRINT(11),TV1(10)
      DATA MAXHC/101/,MAXNOX/50/,TGRID/100./,CGRID/50./
C
      NT=NT1-1
      TCI=FLOAT(JSTRT)
      DO 100 L=1,NOZ
      IF (OZM(L).LE.0.) GO TO 100
C
C SET NORMALIZATION FACTORS AND VERTICAL LABELS
C
      CLOW=0.
      TLOW=0.
      CH=OZM(L)*1.1
      CH1=ALOG10(CH)
      IAMEG=IFIX(CH1)
      IF (CH1.LT.0.) IAMEG=IAMEG-1
      CHIGH=CH/(10.**IAMEG)
      REM=CHIGH-FLOAT(IFIX(CHIGH))
      CHIGH=CH+(1.-REM)*(10.**IAMEG)
      CSPAN=CGRID/CHIGH
      DO 10 I=1,10
      M=11-I
      TV1(I)=(FLOAT(M)*.1)*CHIGH
   10 CONTINUE
C
C SET HORIZONTAL LABELS
C
   20 THIGH=STOPP
   30 TSPAN=TGRID/THIGH
```

```
      DO 40 J=1,11
      TPRINT(J)=(FLOAT(J-1)/10.)*THIGH+.5
      TPRINT(J)=CLOCK(TCI,IFIX(TPRINT(J)))
   40 CONTINUE
C
C CLEAR GRID
C
      MAXHC1=MAXHC-1
      DO 50 J=1,MAXHC
      JSAV(J)=0
   50 CONTINUE
C
      IF (NT.LT.1) NT=1
      DO 60 J=1,NT
      KHC=IFIX((SAVTIM(J)-TLOW)*TSPAN+1.5)
      KNOX=IFIX((SAVCON(J,L)-CLOW)*CSPAN-0.5)
      KNOX=MAXNOX-KNOX
      IF (KNOX.LT.1) GO TO 60
      IF (KNOX.GT.MAXNOX) GO TO 60
      IF (KHC.LT.1) GO TO 60
      IF (KHC.GT.MAXHC1) GO TO 60
      IF (KHC.LT.2) GO TO 60
      JSAV(KHC)=KNOX
   60 CONTINUE
C
      WRITE (IOUT,110) TVERT(1,1),TV1(1)
      DO 90 K=2,MAXNOX
      M=MOD((K-1),5)
      I=(K-1)/5+1
      DO 70 J=2,MAXHC1
      JGRID(J)=JBLANK
      IF (JSAV(J).EQ.K) JGRID(J)=JSYMB
   70 CONTINUE
C
      IF (M.NE.0) GO TO 80
      JGRID(1)=JPLUS
      JGRID(101)=JPLUS
      WRITE (IOUT,130) TVERT(K,1),TV1(I),(JGRID(J),J=1,MAXHC)
      GO TO 90
   80 JGRID(1)=JBAR
      JGRID(101)=JBAR
      WRITE (IOUT,140) TVERT(K,1),(JGRID(J),J=1,MAXHC)
   90 CONTINUE
      WRITE (IOUT,150) CLOW
      WRITE (IOUT,170) TPRINT
      WRITE (IOUT,160) (ITTL(I),I=1,36)
      WRITE (IOUT,180) PLSP(L)
  100 CONTINUE
C
      RETURN
```

```
C                                                                       AM  101
C  FORMAT STATEMENTS                                                    AM  102
C                                                                       AM  103
  110 FORMAT (1H1////9X,A4,F5.3,1H+,10(10H---------+))                  AM  104
  120 FORMAT (1H1//////18X,1H+,10(10H---------+)/18X,101A1/18X,101A1)   AM  105
  130 FORMAT (9X,A4,F5.3,101A1)                                         AM  106
  140 FORMAT (9X,A4,5X,101A1)                                           AM  107
  150 FORMAT (13X,F5.3,1H+,10(10H---------+))                           AM  108
  160 FORMAT (46X,36A2)                                                 AM  109
  170 FORMAT (F21.0,10F10.0/63X,14H  TIME (LDT)  //)                    AM  110
  180 FORMAT (1H0/45X,A4,36H CONCENTRATION AS A FUNCTION OF TIME)       AM  111
      END                                                               AM  112-
      SUBROUTINE CURV1 (N,X,Y,SLP1,SLPN,YP,TEMP,SIGMA)
C
C     THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C     COMPUTE AN INTERPOLATORY SPLINE UNDER TENSION THROUGH
C     A SEQUENCE OF FUNCTIONAL VALUES.  THE SLOPES AT THE TWO
C     ENDS OF THE CURVE MAY BE SPECIFIED OR OMITTED.  FOR ACTUAL
C     COMPUTATION OF POINTS ON THE CURVE IT IS NECESSARY TO CALL
C     THE FUNCTION CURV2.
C
C     ON INPUT --
C     N = THE NUMBER OF VALUES TO BE INTERPOLATED (N.GE.2),
C     X = AN ARRAY OF THE N INCREASING ABCISSAE OF THE
C         FUNCTIONAL VALUES,
C     Y = AN ARRAY OF THE N ORDINATES OF THE VALUES, (I.E.Y(K)
C         IS THE FUNCTIONAL VALUE CORRESPONDING TO X(K)),
C     SLP1,SLPN CONTAIN THE DESIRED VALUES FOR THE FIRST
C         DERIVATIVE TO THE CURVE AT X(1) AND X(N), RESPECTIVELY,
C         IF THE QUANTITY SIGMA IS NEGATIVE THESE VALUES WILL BE
C         DETERMINED INTERNALLY AND THE USER NEED ONLY FURNISH
C         PLACE-HOLDING PARAMETERS FOR SLP1 AND SLPN.  SUCH PLACE-
C         HOLDING PARAMETERS WILL BE IGNORED BY NOT DESTROYED,
C     YP = AN ARRAY OF LENGTH AT LEAST N
C     TEMP = AN ARRAY OF LENGTH AT LEAST N WHICH IS USED FOR
C         SCRATCH STORAGE,
C     SIGMA = THE TENSION FACTOR.  THIS IS NON-ZERO AND
C         INDICATES THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY
C         ZERO (E.G. .001) THE RESULTING CURVE IS APPROXIMATELY A
C         CUBIC SPLINE.  IF ABS(SIGMA) IS LARGE (E.G. 50.) THE
C         RESULTING CURVE IS NEARLY A POLYGONAL LINE.  THE SIGN
C         OF SIGMA INDICATES WHETHER THE DERIVATIVE INFORMATION
C         HAS BEEN INPUT OR NOT.  IF SIGMA IS NEGATIVE THE ENDPOINT
C         DERIVATIVES WILL BE DETERMINED INTERNALLY.  A STANDARD
C         VALUE FOR SIGMA IS APPROXIMATELY 1. IN ABSOLUTE VALUE
C
C     ON OUTPUT --
C     YP = VALUES PROPORTIONAL TO THE SECOND DERIVATIVE
C         OF THE CURVE AT THE GIVEN NODES.
C     N,X,Y,SLP1 AND SIGMA ARE UNALTERED,
C
C      *** AK CLINE/NCAR, COMM. ACM 17,4(APR.1974), 221.
C
      SAVE
      DIMENSION X(N), Y(N), YP(N), TEMP(N)
      COMMON /EXPVAL/ EXPMAX
C
      NCYC=0
      NCHG=0
      NM1=N-1
      NP1=N+1
      DELX1=X(2)-X(1)
      IF (ABS(DELX1/X(2)).LT.0.02) GO TO 110
      DX1=(Y(2)-Y(1))/DELX1
C
C     *** DETERMINE SLOPES IF NECESSARY
      IF (SIGMA.LT.0.) GO TO 90
      SLPP1=SLP1
      SLPPN=SLPN
C
C     *** DENORMALIZE TENSION FACTOR
   10 IF (ABS(X(N)-X(1))/X(N).LT.0.02) GO TO 110
      SIGMAP=ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
      DX1=(Y(2)-Y(1))/DELX1
C
C     *** SET UP RIGHT HAND SIDE OF TRIDIAGONAL SYSTEM
C         FOR YP AND PERFORM FORWARD ELIMINATION
      DELS=SIGMAP*DELX1
      IF (ABS(DELS).GE.EXPMAX) GO TO 110
      EXPS=EXP(DELS)
      SINHS=0.5*(EXPS-1./EXPS)
      SINHIN=1./(DELX1*SINHS)
      DIAG1=SINHIN*(DELS*0.5*(EXPS+1./EXPS)-SINHS)
      DIAGIN=1./DIAG1
      YP(1)=DIAGIN*(DX1-SLPP1)
      SPDIAG=SINHIN*(SINHS-DELS)
      TEMP(1)=DIAGIN*SPDIAG
      IF (N.EQ.2) GO TO 30
      DO 20 I=2,NM1
      DELX2=X(I+1)-X(I)
      IF (ABS(DELX2)/X(I+1).LT.0.02) GO TO 110
      DX2=(Y(I+1)-Y(I))/DELX2
      DELS=SIGMAP*DELX2
      IF (ABS(DELS).GE.EXPMAX) GO TO 110
      EXPS=EXP(DELS)
      SINHS=.5*(EXPS-1./EXPS)
      SINHIN=1./(DELX2*SINHS)
      DIAG2=SINHIN*(DELS*(.5*(EXPS+1./EXPS))-SINHS)
      DIAGIN=1./(DIAG1+DIAG2-SPDIAG*TEMP(I-1))
      YP(I)=DIAGIN*(DX2-DX1-SPDIAG*YP(I-1))
      SPDIAG=SINHIN*(SINHS-DELS)
      TEMP(I)=DIAGIN*SPDIAG
      DX1=DX2
      DIAG1=DIAG2
   20 CONTINUE
   30 DIAGIN=1./(DIAG1-SPDIAG*TEMP(NM1))
      YP(N)=DIAGIN*(SLPPN-DX2-SPDIAG*YP(NM1))
C
C     ***  PERFORM BACK SUBSTITUTION
      DO 40 I=2,N
      IBAK=NP1-I
      YP(IBAK)=YP(IBAK)-TEMP(IBAK)*YP(IBAK+1)
   40 CONTINUE
      IF (SIGMA.GT.O.) RETURN
      IF (N.EQ.3) RETURN
      IF (NCYC.GT.15) RETURN
      IF (NCYC.GT.0) GO TO 50
      XTEST=(X(3)+X(2))/2.
      YCORD=(Y(3)+Y(2))/2.
      FACT=SQRT(ABS(SIGMA))
      YTEST=(YCORD*FACT+CIRC(XTEST,X(1),Y(1))+CIRC(XTEST,X(2),Y(2)))/(2.
     1+FACT)
      ITHH=1
      DY2=CURV2(XTEST,N,X,Y,YP,SIGMA,ITHH)-YTEST
      DSAV=DY2
      SN=-1.
      IF (DY2.GT.O.) SN=1.
      FACT=0.3*(SN+2.)
      DY2=0.
      IT=1
   50 NCYC=NCYC+1
      DY1=DY2
      DY2=CURV2(XTEST,N,X,Y,YP,SIGMA,IT)-YTEST
      IF (NCYC.GT.15) RETURN
      IF (NCYC.GT.13.AND.NCHG.LT.2) GO TO 80
      IF (SIGN(1.,DY1).NE.SIGN(1.,DY2)) GO TO 60
      IF (ABS(DY2).LT.ABS(DY1)) GO TO 70
   60 SN=-SN
      NCHG=NCHG+1
      FACT=SN*((1./(1.-SN*FACT))**(1./3.)-1.)
   70 SLPP1=SLPP1*(1.+SN*FACT)
      GO TO 10
C
   80 IF (DY2/DSAV.LT.0.2) GO TO 70
      SLPP1=SLPPX
      NCYC=15
      GO TO 10
   90 IF (N.EQ.2) GO TO 100
C
C     *** IF NO DERIVATIVES ARE GIVEN USE SECOND ORDER POLYNOMIAL
C         INTERPOLATION ON INPUT DATA FOR VALUES AT ENDPOINTS.
      SLPP1=(Y(2)-Y(1))/(X(2)-X(1))
      DELN=X(N)-X(NM1)
      DELNM1=X(NM1)-X(N-2)
      DELNN=X(N)-X(N-2)
      IF (ABS(DELN)/X(N).LT.0.02) GO TO 110
      IF (ABS(DELN)/X(N).LT.0.02) GO TO 110
      IF (ABS(DELNM1)/X(NM1).LT.0.02) GO TO 110
      IF (ABS(DELNN)/X(N).LT.0.02) GO TO 110
      C1=(DELNN+DELN)/DELNN/DELN
      C2=-DELNN/DELN/DELNM1
      C3=DELN/DELNN/DELNM1
      SLPPN=C3*Y(N-2)+C2*Y(NM1)+C1*Y(N)
      DX2=X(3)-X(2)
      DX31=X(3)-X(1)
      IF (ABS(DX2)/X(3).LT.0.02) GO TO 110
      IF (ABS(DX31)/X(3).LT.0.02) GO TO 110
      C1=-(DX31+DELX1)/DX31/DELX1
      C2=DX31/DELX1/DX2
      C3=-DELX1/DX31/DX2
      SLPPX=C1*Y(1)+C2*Y(2)+C3*Y(3)
      IF (Y(2).GT.Y(1)) SLPPX=AMAX1(0.,SLPPX)
      FACT2=SQRT(ABS(SIGMA))
      SLPPX=(SLPPX+FACT2*SLPP1)/(1.+FACT2)
      SLPP1=SLPPX
      IF (Y(N).GT.Y(NM1)) SLPPN=AMAX1(0.,SLPPN)
      SLPNL=(Y(N)-Y(NM1))/DELN
      SLPPN=(SLPPN+SLPNL*FACT2)/(1.+FACT2)
      SLPPN=(SLPPN-SLPNL*SIGMA)/(1.-SIGMA)
      GO TO 10
C
C     *** IF ONLY TWO POINTS AND NO DERIVATIVES ARE GIVEN, USE
C         STRAIGHT LINE FOR CURVE
C
  100 YP(1)=0.
      YP(2)=0.
      RETURN
  110 SIGMA=-50.
      RETURN
      END
      FUNCTION CURV2 (T,N,X,Y,YP,SIGMA,IT)
C
C    THIS FUNCTION INTERPOLATES A CURVE AT A GIVEN POINT
C    USING A SPLINE UNDER TENSION.  THE SUBROUTINE CURV1 SHOULD
C    BE CALLED EARLIER TO DETERMINE CERTAIN NECESSARY
C    PARAMETERS.
C
C    ON INPUT--
C    T = A REAL VALUE TO BE MAPPED ONTO THE
C        INTERPOLATING CURVE.
C    N = THE NUMBER OF POINTS WHICH WERE INTERPOLATED
C        TO DETERMINE THE CURVE,
C    X,Y = ARRAYS CONTAINING THE ORDINATES AND ABCISSAS
C        OF THE INTERPOLATED POINTS,
C    YP = AN ARRAY WITH VALUES PROPORTIONAL TO THE SECOND
C        DERIVATIVE OF THE CURVE AT THE NODES
C    SIGMA = THE TENSION FACTOR (ITS SIGN IS IGNORED)
C        IT IS AN INTEGER SWITCH.  IF IT IS NOT 1 THIS INDICATES
C        THAT THE FUNCTION HAS BEEN CALLED PREVIOUSLY (WITH N,X,
C        Y,YP, AND SIGMA UNALTERED) AND THAT THIS VALUE OF T
C        EXCEEDS THE PREVIOUS VALUE.  WITH SUCH INFORMATION THE
C        FUNCTION IS ABLE TO PERFORM THE INTERPOLATION MUCH MORE
C        RAPIDLY.  IF A USER SEEKS TO INTERPOLATE AT A SEQUENCE
C        OF POINTS, EFFICIENCY IS GAINED BY ORDERING THE VALUES
C        INCREASING AND SETTING IT TO THE INDEX OF THE CALL.
C        IF IT IS 1 THE SEARCH FOR THE INTERVAL (X(K),X(K+1))
C        CONTAINING T STARTS WITH K=1.
C        THE PARAMETERS N,X,Y,YP AND SIGMA SHOULD BE INPUT
C        UNALTERED FROM THE OUTPUT OF CURV1.
C
C    ON OUTPUT--
C    CURV2 = THE INTERPOLATED VALUE.  FOR T LESS THAN
C        X(1) CURV2 = Y(1).  FOR T GREATER THAN X(N) CURV2 = Y(N).
C
C    NONE OF THE INPUT PARAMETERS ARE ALTERED.
C
C     *** AK CLINE/NCAR, COMM. ACM 17,4(APR.1974), 221
C
      SAVE
      DIMENSION X(N), Y(N), YP(N)
      COMMON /EXPVAL/ EXPMAX
C
      S=X(N)-X(1)
      IT=IABS(IT)
C
C     *** DENORMALIZE SIGMA
      SIGMAP=ABS(SIGMA)*FLOAT(N-1)/S
C
C     *** IF IT.NE. 1 START SEARCH WHERE PREVIOUSLY TERMINATED,
C         OTHERWISE START SEARCH FROM BEGINNING
```

```
      IF (IT.EQ.1) I1=2
C
C     *** SEARCH FOR INTERVAL
   10 DO 20 I=I1,N
      IF (X(I)-T) 20,20,30
   20 CONTINUE
      I=N
C
C     *** CHECK TO INSURE CORRECT INTERVAL
   30 IF (X(I-1).LE.T.OR.T.LE.X(1)) GO TO 40
C
C     *** RESTART SEARCH AND RESET I1
C         (INPUT (IT( WAS INCORRECT)
      I1=2
      GO TO 10
C
C     *** SET UP AND PERFORM INTERPOLATION
   40 IF (SIGMA.LE.(-49.)) GO TO 50
      DEL1=T-X(I-1)
      DEL2=X(I)-T
      DELS=X(I)-X(I-1)
      EXPTST=SIGMAP*DEL1
      IF (ABS(EXPTST).GT.EXPMAX.AND.EXPTST.LT.0.) EXPTST=-EXPMAX
      IF (ABS(EXPTST).GT.EXPMAX.AND.EXPTST.GT.0.) EXPTST=EXPMAX
      EXPS1=EXP(EXPTST)
      SINHD1=.5*(EXPS1-1./EXPS1)
      EXPTST=SIGMAP*DEL2
      IF (ABS(EXPTST).GT.EXPMAX.AND.EXPTST.LT.0.) EXPTST=-EXPMAX
      IF (ABS(EXPTST).GT.EXPMAX.AND.EXPTST.GT.0.) EXPTST=EXPMAX
      EXPS=EXP(EXPTST)
      SINHD2=.5*(EXPS-1./EXPS)
      ALN1=ALOG(ABS(EXPS1))
      ALN2=ALOG(ABS(EXPS))
      IF (ABS(ALN1+ALN2).GT.EXPMAX.AND.EXPS.GT.1.E-10) EXPS=EXP(EXPMAX)
      IF (ABS(ALN1+ALN2).GT.EXPMAX.AND.EXPS.LT.1.E-10)
     1      EXPS=EXP(-EXPMAX)
      IF (ABS(ALN1+ALN2).LT.EXPMAX) EXPS=EXPS1*EXPS
      SINHS=.5*(EXPS-1./EXPS)
      CURV2=(YP(I)*SINHD1+YP(I-1)*SINHD2)/SINHS+((Y(I)-YP(I))*DEL1+(Y(I-
     11)-YP(I-1))*DEL2)/DELS
      IF ((ABS(Y(I)-CURV2)+ABS(Y(I-1)-CURV2)).GT.1.001*(ABS(Y(I)-Y(I-1))
     1)) IT=-IT
      I1=I
      RETURN
   50 IF (I.EQ.1) I=2
      IF (ABS(X(I)-X(I-1))/X(I).LT.0.02) I=I+1
      IF (I.GT.N) I=N
      CURV2=((T-X(I-1))*(Y(I)-Y(I-1))/(X(I)-X(I-1)))+Y(I-1)
      IT=1
      RETURN
      END
      SUBROUTINE RLINE (NC,F,KS1,JS)
C
C  THIS ROUTINE CONTROLS THE ISOPLETH OPTION OF OZIPM
C
      SAVE
      COMMON /NEED/ HC,XN,NL,OZP(20),OZN(11,11,5),MR,LS,HCS,XNS
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSTRT,DTIM,Z1,Z2,DCON,EHC,EXN,F
     1LST,TLST
      COMMON /HOUR/ OZM(5),NGO,TM,DUMMM(5)
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      INTEGER HCSPEC,REACT,PLSP
C
      IF (KS1.GE.121) RETURN
      IF (INFO.EQ.0) WRITE (IOUT,365) (PLSP(I),I=1,NOZ)
      IF (INFO.LT.0) WRITE (IOUT,370) (PLSP(I),I=1,NOZ)
      IF (KS1.GT.0) IBEG=KS1+1
      IF (JS.LE.0) IBEG=1
      INX=1
      DO 20 K=IBEG,121
      I=MOD(K,11)
      IF (I.EQ.0) I=11
      J=(K-I)/11+1
      HCC = FLOAT(J-1)*.1 * HC
      XNN = FLOAT(I-1)*.1 * XN
      CALL SIM (HCC,XNN,OZN(I,J,1),INX)
      DO 10 L=2,5
      OZN(I,J,L)=OZM(L)
   10 CONTINUE
   20 CONTINUE
      RETURN
C
  365 FORMAT (1H1,//,14X,36HTHE FOLLOWING SIMULATIONS WERE DONE./1HO/1H
     1,6X,4HNMOC,11X,3HNOX,12X,5HRATIO,10X,5(A4,11X))
  370 FORMAT (1H1,//,14X,36HTHE FOLLOWING SIMULATIONS WERE DONE./1HO/1H
     1,6X,4HNMOC,5X,3HNOX,7X,5HRATIO,5(6X,A4,6X,4HTIME))
      END
      SUBROUTINE LINER (SIG,KALCMP,F,OZN)
C
C  THIS SUBROUTINE PERFORMS THE ACTUAL PLOTTING OF THE ISOPLETHS
C
      SAVE
      COMMON /NEED/ HC,XN,NL,OZP(20),OZ1(11,11,5),MR,LS,HCS,XNS
      COMMON /VVLB1/ FACTOR,DISTNC,CHRSZ,NCHR,OZLBL
      COMMON /PLTVEC/ HCT(20),OT(20),NT,OHC,HCG,PLTGRD,OXN,XNG,HC1,XN1,T
     1ICZ,DIGZ,CHRZ
      COMMON /CNTRL/ SGG,SIGMA,INFO,NPTO,TSTRT,DTIM,Z1,Z2,DCON,EHC,EXN,F
     1LST,TLST
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /TITL/ ITTL(36)
      COMMON /GEAR6/ SCR(11,11),RXX(9,11),RPLV(20,11),RPLD(20,21),
     1 NRL(20),NSL(20),NBL(20),NTL(20),NSLV(20),NRLV(20),NBLV(20),
     2 NTLV(20),NSLD(20),NRLD(20),NBLD(20),NTLD(20),XNOX(11),XHC(11),
     3W(22),WW(22),OE(22),HCTP(20),XNE(22),XNEP(20),YP(20),TM(20),
     4XNAP(20),HCAP(20),HCLP(20),ITTL2(36),U(88),V(88),HCBP(20),
     5DUM1(378),NDM1(20),NDM2(20)
      COMMON /SCRAT3/ WY(20),ISPN
      DIMENSION F(20), RBL(20,11), RBLV(20,11), RBLD(20,21), ITTL1(36),
     1 OZN(11,11)
      DIMENSION LBLTOP(6),LBLBOT(6),LBLRGT(6),LBLLFT(6)
      DATA LBLTOP,LBLRGT/12*'    '/
      DATA LBLLFT/'NOX ','(PPM',')   ',3*'    '/
      DATA LBLBOT/'NMOC','(PPM','C)  ',3*'    '/,JBL/'  '/
C
C  FIRST SET CALCOMP OPTION
C
      LS=1
      CHRSZ=TICZ
      IDIG1=1
      IDIG2=2
      IHY=IFIX(HC*10.+0.5)
      IHZ=IFIX(XN*1000.+0.5)
      IF (MOD(IHY,10).NE.0) IDIG1=2
      IF (MOD(IHZ,7).NE.0.OR.IHZ.LT.10) IDIG2=3
      HCC=HC1/HC
      HCX=0.
      HCM=HC/10.
      XNC=XN1/XN
      XNX=0.
      XNM=XN/7.
      XSIZE=HC1*.1
      YSIZE=XN1*.1
      XX=((10.0-(HCC*HC))/2.0)+0.3
      YY=((8.0-(XNC*XN))/2.0)+0.3
      ORGY=-.75
      IF (KALCMP.LE.0) GO TO 10
      CALL PLOTS (I1,I2,14)
      CALL PLOT (XX-0.5*XSIZE,YY-0.5*YSIZE,-3)
   10 CONTINUE
      DO 30 J=1,11
      DO 20 I=1,11
      SCR(I,J)=OZN(J,I)
   20 CONTINUE
   30 CONTINUE
      CALL ISOPLT (HCTP,XN,1)
C
C    DRAW THE ISOLINES
C
      CALL VVMNMX (SCR,11,11,11,OMIN,OMAX)
      NW=0
      DO 40 I=1,NL
      IF (OZP(I).LT.OMIN.OR.OZP(I).GT.OMAX) GO TO 40
      NW=NW+1
      WY(NW)=OZP(I)
   40 CONTINUE
      NCHR=4
      NCNT=0
   50 VAL=WY(1)*(10.**NCNT)
      IF (VAL.GT.0.01) GO TO 60
      NCNT=NCNT+1
      GO TO 50
   60 NCHR=NCHR+NCNT
      REMB=WY(1)-FLOAT(IFIX(WY(1)*(10.**(NCNT+2))+0.1))/(10.**(NCNT+2))
      IF (REMB.GE.(0.001/(10.**NCNT))) NCHR=NCHR+1
      IDG1=-(1000*NCHR+NW)
      CALL CONMAP(SCR,11,11,11,XSIZE,YSIZE,WY,IDG1,KALCMP)
C
      IF (KALCMP.LE.0) GO TO 130
      CALL PLOT (0.5*XSIZE,0.5*YSIZE,-3)
      CALL FRAME (0.,0.,0.,HC,HCC,HCX,HCM,IDIG1,0.,XN,XNC,XNX,XNM,IDIG2,
     1LBLBOT,12,LBLLFT,12,LBLTOP,1,LBLRGT,1)
      NCNT=0
      DO 70 I=1,36
      IF (ITTL(I).NE.JBL) GO TO 80
      NCNT=NCNT+1
   70 CONTINUE
   80 NLST=36-NCNT
      DO 90 I=1,NLST
   90 ITTL1(I)=ITTL(I+NCNT)
      NCNT1=36
      DO 100 I=1,36
      II=37-I
      IF (ITTL1(II).NE.JBL) GO TO 110
      NCNT1=NCNT1-1
  100 CONTINUE
  110 NCNT1=NCNT1-NCNT
      ORGX=((HCC*HC)-(FLOAT(NCNT1)*4.*CHRZ))/2.0
      DO 120 I=1,NCNT1                                                  AQ  101
      CALL SYMBOL (ORGX,ORGY,CHRZ,ITTL1(I),0.,2)                        AQ  102
      ORGX=ORGX+2.0*CHRZ                                                AQ  103
  120 CONTINUE                                                          AQ  104
      CALL PLOT (10.,2.,999)                                            AQ  105
  130 HCS=FLOAT(NW)                                                     AQ  106
      XNS=FLOAT(NLL)                                                    AQ  107
      CALL ISOPLT (HC,XN,3)                                             AQ  108
      RETURN                                                            AQ  109
      END                                                               AQ  110-
      SUBROUTINE EDGMX (X,Y,N,XMX,YMX,L)
C
C  CALCULATE THE MAXIMUM POINT ALONG A LINE
C
      SAVE
      DIMENSION X(N), Y(N)
      LL=N
      I=N
      IF (N.LE.2) GO TO 30
      NGO=0
      DO 10 I=2,LL
      IF (Y(I).GT.Y(I-1)) GO TO 10
      IF (I.EQ.LL) L=I
      IF (I.EQ.LL) NGO=1
      IF (I.EQ.LL) GO TO 20
      IF (Y(I).GT.Y(I+1)) L=I+1
      IF (Y(I).GT.Y(I+1)) NGO=1
      GO TO 20
   10 CONTINUE
   20 CONTINUE
      IF (NGO.EQ.1) GO TO 40
   30 L=LL-1
      XMX=X(L+1)
      YMX=Y(L+1)
      RETURN
   40 L=MAX0(L-1,3)
      X21=X(L-1)-X(L-2)
      X221=X(L-1)*X(L-1)-X(L-2)*X(L-2)
      X32=X(L)-X(L-1)
      XL2=X(L)*X(L)
      Y21=Y(L-1)-Y(L-2)
      C=(Y21/X21-(Y(L)-Y(L-1))/X32)/(X221/X21-(XL2-X(L-1)*X(L-1))/X32)
      B=(Y21-C*X221)/X21
      A=Y(L)-B*X(L)-C*XL2
      XMX=-B*0.5/C
      YMX=A+B*XMX+C*XMX*XMX
      IF (XMX.GE.0.999*X(L)) YMX=Y(L)
      IF (YMX.EQ.Y(L)) XMX=X(L)
      IF (XMX.LT.1.001*X(L-2).AND.L.GT.3) GO TO 40
      L=L-1
      IF (XMX.LE.X(L)) L=L-1
      RETURN
      END
      SUBROUTINE ISOPLT (SAVHC,SAVNOX,IENTRY)
C
C  DRAWS A LINE PRINTER PLOT OF THE ISOPLETH
C
      SAVE
      COMMON /NEED/ HC,XN,NL,OZ1(20),OZN(11,11,5),MR,LS,HCS,XNS
      COMMON /TITL/ ITTL(36)
      COMMON /INOUT/ IN,IOUT,ITAPE,IALN,IALL,INHH,IOZC
      COMMON /CALC/ NR,KR(135,12),A(135),S(135),R(135),ITYPE(135),IA(60)
     1,JA(800),DILUT,TEMP,ERR,START,STOPP,SPECIS(61),TPRNT,TSTEP,ZENI
      COMMON /PHOTON/ JGRID(101,42),SPRSE(13)
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /SPEC/ NS,HCSPEC(20),CARB(20),RCTY(20),XNF(2),IH(20),
     1INOX(2),FINHC(20),FALHC(20),NHC,OZIN,OZAL,HCIN,HCAL,XNIN,XNAL,
     2NOZ,FENX(2),C(61),REACT(61),NI,PLSP(5),KOZ(5)
      COMMON /GEAR6/ DUM1(880),XZP(20),DUM2(326),NGRID(101,2),DUM3(572)
      COMMON /HOUR/ OZM(5),NGO,TM,DUMMM(5)
      COMMON /SCRAT3/ OZP(20),ISPN
      DIMENSION TVERT(52,2), TPRINT(11), NLINE(3)
      DIMENSION TV(7), SAVDAT(101), TV1(7)
      INTEGER JBLANK, TV, TVERT
      DATA JBLANK/'    '/,MAXHC/101/,MAXNOX/42/,TGRID/100./,
     1CGRID/42./,JPLUS/'+'/,JBAR/'I'/,JSYMB/'+'/,JBL/' '/
      DATA TV/' N  ',' O  ',' X  ','    ',' P  ',' P  ',' M  '/
      DATA TVERT/104*'    '/
C
C     EMULATE MULTIPLE ENTRY POINTS WITH COMPUTED GO TO
      GO TO (10,90,120),IENTRY
      STOP
   10 CONTINUE
      NR1=1
      DO 20 J=1,42
      IF (J.LE.2) NGRID(1,J)=JBAR
      IF (J.LE.2) NGRID(101,J)=JBAR
      JGRID(1,J)=JBAR
      JGRID(101,J)=JBAR
   20 CONTINUE
      DO 30 I=1,52
      TVERT(I,1)=JBLANK
      IF (I.LT.17.OR.I.GT.23) GO TO 30
      K=I-16
      TVERT(I,1)=TV(K)
   30 CONTINUE
      DO 40 I=1,101
   40 SAVDAT(I)=0.
      DO 50 J=1,42,3
      JGRID(1,J)=JPLUS
      JGRID(101,J)=JPLUS
   50 CONTINUE
```

```
C
C SET NORMALIZATION FACTORS AND VERTICAL LABELS
C
      CLOW=0.
      TLOW=0.
      CHIGH=XN
      CSPAN=CGRID/CHIGH
      THIGH=HC
      DO 60 I=1,7
      M=8-I
      TV1(I)=(FLOAT(M)/7.)*CHIGH
   60 CONTINUE
C
C SET HORIZONTAL LABELS
C
      TSPAN=TGRID/THIGH
      DO 70 J=1,11
      TPRINT(J)=(FLOAT(J-1)/10.)*THIGH
   70 CONTINUE
C
C CLEAR GRID
C
      MAXHC1=MAXHC-1
      DO 80 K=1,MAXNOX
      DO 80 J=2,MAXHC1
      JGRID(J,K)=JBLANK
      IF (K.LE.2) NGRID(J,K)=JBLANK
   80 CONTINUE
      RETURN
C
C ENTRY FOR SAVING INTERPOLATED POINTS
C
C     ENTRY SAVLIN
   90 CONTINUE
      KHC=IFIX((SAVHC-TLOW)*TSPAN+1.5)
      KNOX=IFIX((SAVNOX-CLOW)*CSPAN-0.5)
      KNOX=MAXNOX-KNOX
      IF (KNOX.LT.1) GO TO 110
      IF (KNOX.GT.MAXNOX) GO TO 110
      IF (KHC.LT.1) GO TO 110
      IF (KHC.GT.MAXHC1) GO TO 110
      JGRID(KHC,KNOX)=JSYMB
      IF (TM.EQ.1..AND.NPTO.EQ.0) DUM3(NR1)=FLOAT(KHC)
      IF (TM.EQ.1.) NR1=NR1+1
      IF (KHC.LT.2) GO TO 110
  100 CONTINUE
  110 RETURN
C
C ENTRY FOR PLOTTING GRID
C
```

```
C     ENTRY LINPRT
  120 CONTINUE
      NPL=IFIX(HCS+0.1)
      NLL=IFIX(XNS+0.5)
      KOUNT=0
      IF (NLL.EQ.0) GO TO 160
      NJ=1
  130 NJ=NJ+1
      J=MAXNOX-NJ
      IF (JGRID(3,J).EQ.JBLANK) GO TO 130
      KOUNT=KOUNT+1
      IF (KOUNT.GT.NLL) GO TO 150
      IF (NJ.GT.MAXNOX-1) GO TO 150
      NVAR=IFIX(OZP(KOUNT)*100.+0.5)
      CALL CONVT (NVAR,NLINE,3)
      DO 140 II=1,3
         II2=II+2
  140 IF (NLINE(II).NE.JBL) JGRID(II2,J)=NLINE(II)
      NJ=NJ+1
      GO TO 130
  150 KOUNT=KOUNT-1
  160 KOUNT=KOUNT+1
      NR1=NR1-1
      DO 180 I=KOUNT,NR1
      NVAR=IFIX(OZP(I)*100.+0.5)
      CALL CONVT (NVAR,NLINE,3)
      KHC=IFIX(DUM3(I)+0.1)
      DO 170 II=2,3
  170 IF (NLINE(II).NE.JBL) NGRID(KHC,II-1)=NLINE(II)
  180 CONTINUE
      WRITE (IOUT,220) ((NGRID(I,J),I=1,101),J=1,2)
      KFRST=1
      DO 190 K=KFRST,MAXNOX
      L=MOD((K-1),6)
      I=(K-1)/6+1
      IF (L.EQ.0) WRITE (IOUT,230) TVERT(K,1),TV1(I),(JGRID(J,K),J=1,MAX
     1HC)
      IF (L.NE.0) WRITE (IOUT,240) TVERT(K,1),(JGRID(J,K),J=1,MAXHC)
  190 CONTINUE
      WRITE (IOUT,250) CLOW
      WRITE (IOUT,260) TPRINT
      WRITE (IOUT,270) (ITTL(I),I=1,36)
      IF (NPTO.EQ.0) WRITE (IOUT,280) ISPN,(OZP(I),I=1,NPL)
  200 RETURN
C
C
  210 FORMAT (1H1/////////9X,A4,F5.3,1H+,10(10H---------+))
  220 FORMAT (1H1//////18X,1H+,10(10H---------+)/18X,101A1/18X,101A1)
  230 FORMAT (9X,A4,F5.3,101A1)
  240 FORMAT (9X,A4,5X,101A1)
```

```
250 FORMAT (13X,F5.3,1H+,10(10H---------+))                           AS  151
260 FORMAT (F21.3,10F10.2/64X,11HNMOC (PPMC)//)                       AS  152
270 FORMAT (46X,36A2)                                                 AS  153
280 FORMAT (1H0/20X,4HTHE ,A4,12H LINES ARE  ,10F8.5/40X,10F8.5)      AS  154
    END                                                               AS  155-
```

```
      SUBROUTINE VVMNMX (Z,ND1,NX,NY,ZMIN,ZMAX)
C
C     *** RETURNS THE MINIMUM AND MAXIMUM VALUES OF AN ARRAY
C             GWL/SAI  DEC 77
C
      SAVE
      DIMENSION Z(ND1,1)
C
      ZMIN = +1.E20
      ZMAX = -1.E20
C
      DO 20 J=1,NY
          DO 10 I=1,NX
              ZIJ = Z(I,J)
C
              IF (ZIJ .LT. ZMIN) ZMIN = ZIJ
              IF (ZIJ .GT. ZMAX) ZMAX = ZIJ
   10     CONTINUE
   20 CONTINUE
C
      RETURN
      END
      SUBROUTINE CONMAP (Z,ND1,NOX,NOY,SIZX,SIZY,C,NLEV,KALCM1)
      SAVE
C
C     *** PURPOSE -- DRAWS A CONTOUR MAP
C             GW LUNDBERG/SAI  JULY 87
C
C     Z    THE (ORIGIN OF THE) ARRAY TO BE CONTOURED
C     ND1  THE FIRST DIMENSION OF Z
C     NOX  THE NUMBER OF ELEMENTS OF Z TO USE IN X-DIRECTION
C     NOY  THE NUMBER IN THE Y-DIRECTION
C     SIZX INCHES/CELL IN THE X-DIRECTION
C     SIZY INCHES/CELL IN THE Y-DIRECTION
C     C    THE CONTOUR LEVELS TO DRAW
C     NLEV MOD(IABS(NLEV),1000) IS THE NUMBER OF LEVELS
C          IF NLEV<0, THE CONTOURS ARE LABELED WITH
C             -NLEV/1000 DECIMAL DIGITS OF PRECISION
C          IF NLEV>=1000, THE CONTOURS ARE LABELED
C             NLEV/1000, NLEV/1000+1,.....,NLEV/1000+NL-1
C             AS INTEGERS WHERE NL=MOD(NLEV,1000)
C
C     *** NOTE -- MISSING DATA CELLS ARE NOT MAPPED
C
      DIMENSION Z(ND1,1),C(1),LBL(3)
      LOGICAL NOLBL
      COMMON /VVCONM/ IX,JY,IDX,JDY,NBR,LOOP,NP,NTBL(1000),NR,
     1    NX,NY,XSIZ,YSIZ,XXFLG,KALCMP
      COMMON /VVLB1/ FACT,DIST,CHRSZ,NCHR,OZL
      DATA NR1/1000/
      NR=NR1
      KALCMP=KALCM1
C
C     *** MOVE ARGUMENTS TO COMMON
      NX = NOX
      NY = NOY
      XSIZ = SIZX
      YSIZ = SIZY
C
C     *** SET THE NUMBER OF LEVELS REQUESTED
      NC = MOD(IABS(NLEV),1000)
C
C     *** ASSUME NO LABELS
      NOLBL = .TRUE.
C
C     *** SET LABEL SIZE JUST IN CASE
      SIZLBL = 0.07
C     *** SET UP LABELING PARAMETERS AS REQUESTED
      IF (NLEV .GT. 0) GO TO 10
C
      NDEC = -NLEV/1000
      IF (NDEC .EQ. 0) NDEC = -1
      NOLBL = .FALSE.
      GO TO 20
C
   10 IF (NLEV .LT. 1000) GO TO 20
      LOW = NLEV/1000 - 1
      NDEC = -1
      NOLBL = .FALSE.
C
C
C     *** FOR EACH LEVEL, LOCATE THE STARTING POINT OF EACH
C         CONTOUR, THEN CALL VVCMAP TO DRAW THE LINE
   20 DO 120 L=1,NC
C
C         NORMALIZE THE CONTOUR LEVEL
C
          CV = C(L)/2.0
          CV = CV*2.0
C
C         *** SET UP LABELS IF REQUESTED
          IF (NOLBL) GO TO 30
          OZL = CV
C
   30     NP = 0
C
C         *** IN THE FOLLOWING, (IX,JY) IS THE GRID POINT (NODE)
C             JUST HIGHER THAN CV.  NBR IS ONE OF FOUR NEIGHBORING
C             NODES (1=WEST, 3=N, 5=E, 7=S) TO BE USED WITH (IX,JY) TO
C             INTERPOLATE THE COORDINATES OF THE CONTOUR.  IDX AND JDY
C             INDICATE THE DIRECTION OF THIS NEIGHBOR.
C
C         *** SCAN THE FOUR EDGES FOR OPEN CONTOURS.  THIS SCAN CLEVERLY
C             CAPTURES EVERY OPEN CONTOUR EXACTLY ONCE
          LOOP = 0
C
C         *** SCAN BOTTOM AND TOP EDGE
          DO 50 I=2,NX
              IM1 = I-1
              IF (Z(IM1,1) .GE. CV .OR. Z(I,1) .LT. CV) GO TO 40
              IX = I
              JY = 1
              IDX = -1
              JDY = 0
              NBR = 1
              CALL VVCMAP (Z,ND1,CV)
C
   40         IF (Z(I,NY) .GE. CV .OR. Z(IM1,NY) .LT. CV) GO TO 50
              IX = IM1
              JY = NY
              IDX = 1
              JDY = 0
              NBR = 5
              CALL VVCMAP (Z,ND1,CV)
   50     CONTINUE
C
C         *** SCAN RIGHT AND LEFT EDGE
          DO 70 J=2,NY
              JM1 = J-1
              IF (Z(NX,JM1) .GE. CV .OR. Z(NX,J) .LT. CV) GO TO 60
              IX = NX
              JY = J
              IDX = 0
              JDY = -1
              NBR = 7
              CALL VVCMAP (Z,ND1,CV)
C
   60         IF (Z(1,J) .GE. CV .OR. Z(1,JM1) .LT. CV) GO TO 70
              IX = 1
              JY = JM1
              IDX = 0
              JDY = 1
              NBR = 3
              CALL VVCMAP (Z,ND1,CV)
   70     CONTINUE
C
C         *** SCAN CENTER REGION FOR CLOSED LOOPS -- IT IS SUFFICIENT
C             TO CHECK ONLY THE WEST NEIGHBOR AT EACH NODE
          LOOP = -1
          NYM1 = NY-1
          DO 110 J=2,NYM1
              DO 100 I=2,NX
                  IM1 = I-1
                  IF (Z(IM1,J) .GE. CV .OR. Z(I,J) .LT. CV) GO TO 100
C
C                 *** CHECK IF NODE ALREADY USED
                  NODE = 1000*I + J
                  IF (NP .EQ. 0) GO TO 90
                  DO 80 K=1,NP
                      IF (NODE .EQ. NTBL(K)) GO TO 100
   80             CONTINUE
C
C                 *** TAG THIS NODE TO PREVENT RETRACE
   90             NP = NP+1
                  IF (NP .GT. NR) GO TO 120
                  NTBL(NP) = NODE
C
C                 *** SET PARAMETERS AND DRAW LOOP
                  IX = I
                  JY = J
                  IDX = -1
                  JDY = 0
                  NBR = 1
                  CALL VVCMAP (Z,ND1,CV)
  100         CONTINUE
  110     CONTINUE
  120 CONTINUE
C
      RETURN
      END
      SUBROUTINE VVCMAP (Z,ND1,C)
      SAVE
C
C     *** PURPOSE -- TRACES AND DRAWS A CONTOUR GIVEN THE BEGINNING
C         BY CONMAP
C             GW LUNDBERG/SAI  JULY 87
C
C     Z    THE (ORIGIN OF THE) ARRAY TO BE CONTOURED
C     ND1  THE FIRST DIMENSION OF Z
C     C    THE CONTOUR LEVEL TO DRAW
C
      DIMENSION Z(ND1,1),INX(8),JNY(8),UT(11),VT(11),TM(20),YP(20)
      COMMON /VVCONM/ IX,JY,IDX,JDY,NBR,LOOP,NP,NTBL(1000),NR,
     1     NX,NY,XSIZ,YSIZ,XXFLG,KALCMP
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      DATA INX /-1,-1,0,1,1,1,0,-1/
      DATA JNY /0,1,1,1,0,-1,-1,-1/
      DATA FUZZ/0.003/
C
      CV = C
C
C     *** SAVE STARTING NODE AND CURRENT NEIGHBOR
      IXO = IX
      JYO = JY
      NBRO = NBR
C
C     *** SET THE SECOND POINT FOR INTERPOLATION
      IX1 = IX + IDX
      JY1 = JY + JDY
C
C     *** IF EITHER POINT MISSING -- IGNORE WITH VVKURV MODE RESET
      MODE = -1
      IF (Z(IX,JY) .EQ. XXFLG .OR. Z(IX1,JY1) .EQ. XXFLG)
     1     GO TO 30
C
C     *** INTERPOLATE FOR THE FIRST POINT AND SAVE
      IF (IDX .EQ. 0) GO TO 10
      Y = JY
      X = (Z(IX,JY)-CV)/(Z(IX,JY)-Z(IX1,JY))*IDX + IX
      GO TO 20
C
   10 X = IX
      Y = (Z(IX,JY)-CV)/(Z(IX,JY)-Z(IX,JY1))*JDY + JY
C
C     *** SCALE AND SEND TO SPLINE ROUTINE
   20 X = (X-0.5)*XSIZ
      Y = (Y-0.5)*YSIZ
      CALL VVKURV (X,Y,-1)
C
```

```
      MODE = 0
C
C
C     *** MAIN LOOP -- STARTING AT (IX,JY) AND KNOWING THAT THIS
C         NODE IS JUST ON THE HIGH SIDE OF THE CONTOUR, CHECK EACH
C         NEIGHBORING POINT IN THE ORDER W,NW,N,NE,E,SE,S,SW, IF
C         NEIGHBOR IS LOWER THAN CV AND NOT A DIAGONAL -- NW,NE,SE,
C         SW -- INTERPOLATE AND PASS (X,Y) TO THE SPLINE ROUTINE.
C         IF NEIGHBOR IS A DIAGONAL, SIMPLY MOVE ON TO NEXT NEIGHBOR. IF
C         THE NEIGHBOR IS ON HIGH SIDE OF CONTOUR, MOVE (IX,JY)
C         TO THAT POINT AND AGAIN SCAN THE NEIGHBORS STARTING FROM
C         WHERE (IX,JY) WAS (PLUS ONE).
C
C         NOTE -- IF EITHER NODE HAS A Z MISSING VALUE, THE CONTOUR
C         IS TERMINATED ALTHOUGH THE SCAN CONTINUES.  THE CONTOUR
C         REAPPEARS WHEN BOTH NODES ARE AGAIN KNOWN.
   30 NBR = NBR + 1
      IF (NBR .GT. 8) NBR = NBR - 8
      IDX = INX(NBR)
      JDY = JNY(NBR)
      IX1 = IX + IDX
      JY1 = JY + JDY
      IF (LOOP .NE. 0) GO TO 40
C
C     *** CHECK FOR END OF OPEN CONTOUR
      IF (IX1 .GT. NX  .OR.  IX1 .LT. 1  .OR.
     1    JY1 .GT. NY  .OR.  JY1 .LT. 1) GO TO 200
C
C     *** CHECK IF POINT BELOW LEVEL
   40 IF (CV .GT. Z(IX1,JY1)) GO TO 60
C
C     *** REVERSE NODES AND GO ON (UNLESS CELL IS DEGENERATE)
C     IF (MOD(NBR,2) .NE. 0) GO TO 120
C
      NNBR = NBR+1
      IF (NNBR .GT. 8) NNBR = NNBR-8
      NX1 = IX + INX(NNBR)
      NY1 = JY + JNY(NNBR)
C
C     *** BE CAREFUL OF EDGE
      IF (NX1 .GT. NX  .OR.  NX1 .LT. 1  .OR.
     1    NY1 .GT. NY  .OR.  NY1 .LT. 1) GO TO 50
C
C     *** TEST FOR DEGENERACY
      IF (CV .LE. Z(NX1,NY1)) GO TO 50
C
C     *** CELL IS DEGENERATE -- ESTIMATE CENTER HEIGHT
      ZCEN = 0.25 * (Z(IX,JY)+Z(IX,JY1)+Z(IX1,JY1)+Z(IX1,JY))
C
C     *** PUNT IF CENTER LOW
```

```
      IF (CV .GT. ZCEN) GO TO 30
C
   50 NBR = NBR + 4
      IX = IX1
      JY = JY1
      GO TO 30
C
C     *** MAY BE OK -- BUT CHECK FOR MISSING POINT
   60 IF (Z(IX,JY) .NE. XXFLG  .AND.  Z(IX1,JY1) .NE. XXFLG)
     1     GO TO 70
C
C     *** ONE POINT MISSING -- FLUSH SPLINE BUFFER TO SET
C         FOR A BROKEN LINE SEGMENT IF NECESSARY
      IF (MODE .EQ. -1) GO TO 180
      CALL VVKURV (DUM,DUM,2)
      GO TO 180
C
C     *** LOOKS OK -- INTERPOLATE FOR X AND Y
C
C     *** BUT MAY BE A DIAGONAL NODE
   70 IF (MOD(NBR,2) .EQ. 0) GO TO 30
      IF (IDX .EQ. 0) GO TO 120
      Y = JY
C     X = (Z(IX,JY)-CV)/(Z(IX,JY)-Z(IX1,JY))*IDX + IX
C
C  USE SPLINE FIT OF DATA
C
      IF (IDX.GT.0) GO TO 90
      IPTS=IX
      DO 80 JJ=1,IX
      UT(JJ)=Z(JJ,JY)
      VT(JJ)=FLOAT(JJ)
   80 CONTINUE
      GO TO 110
   90 IPTS=12-IX
      DO 100 JJ=1,IPTS
      UT(JJ)=Z(12-JJ,JY)
      VT(JJ)=FLOAT(12-JJ)
  100 CONTINUE
C
  110 SX=-SIG*30.
      IF (IPTS.GT.2) SX=-SIG*20./(FLOAT(IPTS-2)**2)
      CALL CURV1(IPTS,UT,VT,SP1,SP2,YP,TM,SX)
      IT=1
      X=CURV2(CV,IPTS,UT,VT,YP,SX,IT)
C
      GO TO 170
C
  120 X = IX
C     Y = (Z(IX,JY)-CV)/(Z(IX,JY)-Z(IX,JY1))*JDY + JY
```

```
C
C  USE SPLINE FIT OF DATA
C
      IF (JDY.GT.0) GO TO 140
      IPTS=JY
      DO 130 JJ=1,JY
      UT(JJ)=Z(IX,JJ)
      VT(JJ)=FLOAT(JJ)
  130 CONTINUE
      GO TO 160
  140 IPTS=12-JY
      DO 150 JJ=1,IPTS
      UT(JJ)=Z(IX,12-JJ)
      VT(JJ)=FLOAT(12-JJ)
  150 CONTINUE
C
  160 SX=-SIG*30.
      IF (IPTS.GT.2) SX=-SIG*20./(FLOAT(IPTS-2)**2)
      CALL CURV1(IPTS,UT,VT,SP1,SP2,YP,TM,SX)
      IT=1
      Y=CURV2(CV,IPTS,UT,VT,YP,SX,IT)
C
C     *** SCALE AND SEND
  170 X = (X-0.5)*XSIZ
      Y = (Y-0.5)*YSIZ
      CALL VVKURV (X,Y,MODE)
C
C     *** RESET MODE
      MODE = 0
C
C     *** SET NODE IN NTBL SO ITS NOT RETRACED
  180 IF (NBR .NE. 1) GO TO 190
      NP = NP + 1
      IF (NP .GT. NR) GO TO 200
      NTBL(NP) = 1000*IX + JY
C
C     *** CHECK IF LOOP COMPLETE
  190 IF (LOOP .EQ. 0) GO TO 30
      IF (IX .NE. IX0) GO TO 30
      IF (JY .NE. JY0) GO TO 30
      IF (NBR .NE. NBR0) GO TO 30
C
C     *** ALL DONE -- FINISHED OFF THIS LINE AND RETURN
  200 CALL VVKURV (X,Y,1)
      RETURN
      END
      SUBROUTINE VVKURV (XPT,YPT,MODE)
      SAVE
C
C     *** PLOTS COMPLICATED SPLINE FITTED LINE BY SECTIONS
C             GW LUNDBERG/SAI  APR 79
C
C     *** NOTE -- THE POINTS ARE RECEIVED ONE AT A TIME
C
C        XPT    NODE X COORDINATE IN INCHES FROM CURRENT PLOT ORIGIN
C        YPT    NODE Y COORDINATE IN INCHES FROM CURRENT PLOT ORIGIN
C        MODE   <0, INITIAL POINT OF FIRST SECTION
C               =0, ADDITIONAL POINT
C               =1, FLUSH THE BUFFER -- XPT,YPT FORCED
C               =2, FLUSH THE BUFFER -- XPT,YPT IGNORED
C
C     *** NOTE -- THIS CODE PROVIDES FOR BROKEN SPLINED LINES.
C         USE MODE=1 TO TERMINATE THE SUBLINE. A NEW LINE IS
C         STARTED WITH MODE=-1, OR A NEW SUBLINE WITH MODE=0.
C         THIS FEATURE IS USED BY CONMAP FOR MISSING DATA.
C     LOG
C
C     7/17/85 RGJ CHANGED CODE TO SET NPTS TO ZERO IF THERE IS
C     A BROKEN LINE AND NPTS WAS 1.
C
C
C     *** SEE VVKRV1 FOR A DESCRIPTION OF THESE VARIABLES
      COMMON /VVCONM/ IDUM(1010),XSIZ,YSIZ,XXFLG,KALCMP
      COMMON /VVKRV/ NPTS,NSLP,SLP1,SLPN,LSLP,TANGNT,S,
     1     X(32),Y(32),XP(32),YP(32)
      COMMON /VVLB1/ FCTR,DIST,CHRSZ,NCHR,OZLBL
      COMMON /HOUR/ OZR(5),NGG,TM,DUM(5)
      COMMON /NEED/ HC,XN,NL,OZP(20),OZN(11,11,5),MR,LS,HCS,XNS
C
      DATA STEP/0.04/, MXPT/31/, FUZZ/0.05/,NPTS1/0/
      DATA TANN/0./, TINY/0.0001/,LSLP1/0/,IFRST/1/
      IF (IFRST.EQ.1) NPTS=NPTS1
      IF (IFRST.EQ.1) LSLP=LSLP1
      IFRST=2
C
C     *** JUMP TO APPROPRIATE CODE
      IF (MODE) 10,20,60
C
C*****************************************
C         ENTRY FOR MODE LESS THAN ZERO
C*****************************************
C
C     *** INITIAL POINT RECEIVED -- SET SECTION COUNT, INITIALIZE
C         VVLABL AND SAVE THE POINT
   10 NSECTN = 1
      USX=XPT*HC*.1/XSIZ
      USY=YPT*XN*.1/YSIZ
      TM=1.
      CALL ISOPLT(USX,USY,2)
      TM=2.
      IF (KALCMP.GT.0) CALL VVLABL (XPT,YPT,3)
      XSTRT = XPT
      YSTRT = YPT
      X(1) = XPT
      Y(1) = YPT
      NPTS = 1
C
C     *** ZERO KNOWN SLOPE FLAG
      KSLP = 0
      RETURN
C
C*********************************************
C         ENTRY FOR MODE EQUAL ZERO
C*********************************************
C
   20 IF (NPTS .GT. 0) GO TO 30
C
C     *** THIS IS THE CONTINUATION OF A BROKEN LINE (SEE
C         CONMAP) -- SET PEN
      IF (KALCMP.GT.0) CALL VVLABL (XPT,YPT,3)
      GO TO 40
C
C     *** SAVE POINT (IF NOT TOO CLOSE TO LAST POINT)
C         IN BUFFER -- FLUSH WHEN FULL
   30 IF (ABS(X(NPTS)-XPT) .LT. FUZZ  .AND.
     1    ABS(Y(NPTS)-YPT) .LT. FUZZ) RETURN
   40 NPTS = NPTS+1
      X(NPTS) = XPT
      Y(NPTS) = YPT
      IF (NPTS .LT. MXPT) RETURN
C
C     *** FLUSH THE BUFFER -- SUPPLY STARTING SLOPE IF AVAILABLE
      NSLP = KSLP
      SLP1 = TANN
C
      CALL VVKRV1
C
C     *** SET NUMBER OF POINTS TO INTERPOLATE AND NORMALIZED STEP SIZE
      N = S/STEP - 1
      FACT = STEP/S
      DIST=S
C
C     *** SAVE THE SLOPE FOR FIRST SECTION ONLY
      IF (NSECTN .EQ. 1) TAN1 = SLP1
C
C     *** INITIALIZE VVKRV2 (PEN ALREADY THERE)
```

```
      CALL VVKRV2( 0.,XX,YY)
C
C     *** PLOT THE MIDDLE INTERPOLATED POINTS
      DO 50 I=1,N
          T = -FLOAT(I)*FACT
          CALL VVKRV2 (T,XX,YY)
          USX=XX*HC*.1/XSIZ
          USY=YY*XN*.1/YSIZ
          CALL ISOPLT(USX,USY,2)
          IF (KALCMP.LE.0) GO TO 50
          CALL VVLABL (XX,YY,2)
   50 CONTINUE
C
C     *** GET SLOPE AT LAST POINT, SAVE, AND PLOT LAST SEGMENT
      CALL VVKRV2 (-1.,XX,YY)
      TANN = SLPN
      USX=XX*HC*.1/XSIZ
      USY=YY*XN*.1/YSIZ
      CALL ISOPLT(USX,USY,2)
      IF (KALCMP.GT.0) CALL VVLABL (XX,YY,2)
C
C     *** BUMP SECTION COUNT
      NSECTN = NSECTN + 1
C
C     *** MAKE THE LAST POINT THE FIRST FOR THE NEXT SECTION
      X(1) = X(NPTS)
      Y(1) = Y(NPTS)
      NPTS = 1
C
C     *** FLAG FIRST SLOPE AS KNOWN
      KSLP = 1
      RETURN
C
C*********************************************
C         ENTRY FOR MODE GREATER THAN ZERO
C*********************************************
C
C     *** FLUSH THE BUFFER
C
C     *** PUNT IF ONLY FLUSHING
   60 IF (MODE .EQ. 2) GO TO 70
C
C     *** FORCE FINAL POINT IF NOT ALREADY ENTERED
      IF (X(NPTS) .EQ. XPT  .AND.  Y(NPTS) .EQ. YPT) GO TO 70
C
C     *** THERE IS ALWAYS AN EXTRA POSITION IN X AND Y
      NPTS = NPTS+1
      X(NPTS) = XPT
      Y(NPTS) = YPT
C
```

```
C     *** IGNORE SINGLE OR NULL POINT
C
C---  START OF FIX 7/17/87
C
   70 IF (MODE .NE. 2) GO TO 80
      IF (NPTS .EQ. 1) KSPL = 0
      IF (NPTS .LE. 1) NPTS = 0
   80 IF (NPTS .LE. 1) RETURN
C
C---  END OF FIX 7/17/87
C
C     *** CHECK FOR SIMPLE LINE -- ONLY ONE SECTION
      IF (NSECTN .GT. 1) GO TO 110
C
C     ***  TWO POINT LINES TAKE SPECIAL TREATMENT
      IF (NPTS .GT. 2) GO TO 90
C
C     *** TWO POINTS -- DRAW SEGMENT AND RETURN
      USX=X(1)*HC*.1/XSIZ
      USY=Y(1)*XN*.1/YSIZ
      CALL ISOPLT(USX,USY,2)
      USX=X(2)*HC*.1/XSIZ
      USY=Y(2)*XN*.1/YSIZ
      CALL ISOPLT(USX,USY,2)
      IF (KALCMP.LE.0) GO TO 160
      CALL VVLABL (X(1),Y(1),3)
      CALL VVLABL (X(2),Y(2),2)
      CALL VVLABL (DUM,DUM,1)
      GO TO 160
C
C     *** SIMPLE BUT MAY BE A LOOP
   90 IF (X(NPTS) .EQ. XSTRT  .AND.  Y(NPTS) .EQ. YSTRT) GO TO 100
C
C     *** A SIMPLE SIMPLE LINE
      NSLP = 0
      CALL VVKRV1
      GO TO 130
C
C     *** SIMPLE LOOP
  100 NSLP = -1
      CALL VVKRV1
      GO TO 130
C
C     *** COMPLEX LINE AND MAY BE LOOP
  110 IF (X(NPTS) .EQ. XSTRT  .AND.  Y(NPTS) .EQ. YSTRT) GO TO 120
C
C     *** COMPLEX OPEN LINE
      NSLP = 1
      SLP1 = TANN
      CALL VVKRV1
      GO TO 130
C
C     *** COMPLEX LOOP
  120 NSLP = 3
      SLP1 = TANN
      SLPN = TAN1
      CALL VVKRV1
C
C     *** JUST MARK THE POINT IF ARC LENGTH SMALL
  130 IF (S .GT. 0.01) GO TO 140
      IF (KALCMP.LE.0) GO TO 160
      CALL PLOT (XSTRT,YSTRT,3)
      CALL PLOT (XSTRT,YSTRT,2)
      GO TO 160
C
C     *** PLOT THE FINAL SECTION
  140 N = S/STEP - 1.
      FACT = STEP/S
      DIST = S
      CALL VVKRV2 (0.,XX,YY)
      DO 150 I=1,N
          T = -I*FACT
          CALL VVKRV2 (T,XX,YY)
          USX=XX*HC*.1/XSIZ
          USY=YY*XN*.1/YSIZ
          CALL ISOPLT(USX,USY,2)
          IF (KALCMP.LE.0) GO TO 150
          CALL VVLABL (XX,YY,2)
  150 CONTINUE
C
      CALL VVKRV2 (-1.,XX,YY)
      USX=XX*HC*.1/XSIZ
      USY=YY*XN*.1/YSIZ
      CALL ISOPLT(USX,USY,2)
      IF (KALCMP.LE.0) GO TO 160
      CALL VVLABL (XX,YY,2)
      CALL VVLABL (DUM,DUM,1)
C
  160 NPTS = 0
      KSLP = 0
      RETURN
      END
      SUBROUTINE VVLABL (X2,Y2,IPEN)
      SAVE
C
C     *** SETS A LINE LABEL INTO A VECTOR PLOT PROVIDING THAT
C         PARAMETERS IN /VVLBL/
C             GW LUNDBERG/SAI  AUG 87
C
C     X2,Y2    THE TERMINAL POINT OF THE CURRENT VECTOR
C              IN INCHES FROM PRESENT PLOT ORIGIN
C     IPEN     =3, INITIALIZE -- SET PEN TO X2,Y2
C              =2, CONTINUE -- DRAW LINE WITH LABEL TO X2,Y2
C              =1, FINISH -- COMPLETE LINE
C
C     *** NOTES --
C         (1) THE VALUES FOR IPEN MAY APPEAR WHIMSICAL, BUT THEY
C             FOLLOW THE PRECEDENT SET BY SUBROUTINE PLOT.
C         (2) THE LABELS ARE SEPERATED BY DIST INCHES EXCEPT FOR
C             THE FIRST LABEL WHICH STARTS FRST INCHES FROM
C             THE BEGINNING OF THE VECTOR PLOT -- THIS PROVIDES FOR
C             STAGGERED LABELS.
C
      DIMENSION XSV(100),YSV(100)
C
C     *** DIST IS THE DISTANCE IN INCHES BETWEEN LABELS FROM BEGINNING
C         TO BEGINNING.  FRST IS THE DISTANCE TO THE FIRST LABEL.
C         CHRSZ IS THE SIZE OF THE LABEL CHARACTERS IN INCHES.
C         NCHR IS THE NUMBER OF CHARACTERS (0-10), AND LABEL
C         IS THE A-FORMATED TEXT OF THE LABEL
      COMMON /VVLB1/ FACT,DIST,CHRSZ,NCHR,OZL
      DATA MXSV/100/
C
C     *** BRANCH ACCORDING TO IPEN
      GO TO (100,20,10), IPEN
C
C     *** MOVE THE PEN TO THE FIRST POINT
   10 CALL PLOT (X2,Y2,3)
C
C     *** SET NUMBER OF SAVED SEGMENTS TO ZERO
      NSV = 0
C
C     *** IF THERE ARE TO BE NO LABELS -- JUST RETURN
      IF (NCHR .EQ. 0) RETURN
C
C     *** SET UP THE OFFSET NECESSARY TO CENTER THE LABEL AND
C         THE DISTANCE REQUIRED BY THE LABEL
      OFF = CHRSZ/2.
      SZLBL = NCHR*CHRSZ + 2.*OFF - 0.4286*CHRSZ
C
C     *** INITIALIZE THE ACCUMLATED LENGTH OF THE VECTORS AND THE
C         LENGTH REQUIRED BEFORE THE FIRST LABEL.
```

```
      TOTSZ = 0.
      SKPSZ = FACT*DIST
C
C     *** REMEMBER THE STARTING LOCATION OF FIRST VECTOR
      X1 = X2
      Y1 = Y2
C
      RETURN
C
C****************************************************************
C                   ENTRY FOR IPEN = 2
C****************************************************************
C
C     *** IF THERE ARE TO BE NO LABELS -- PLOT THE VECTOR AND RETURN
   20 IF (NCHR .GT. 0) GO TO 30
      CALL PLOT (X2,Y2,2)
      RETURN
C
C     *** IF SEEKING ROOM FOR THE LABEL -- SKIP FOLLOWING
   30 IF (NSV .GT. 0) GO TO 50
C
C     *** CALCULATE THIS VECTOR LENGTH AND ADD TO THE ACCUMULATED
C         LENGTH.  IF A LABEL IS TO START IN THIS VECTOR, SKIP TO
C         120, ELSE PLOT THE VECTOR AND RETURN
      VECSZ = SQRT ((X2-X1)**2 + (Y2-Y1)**2)
      TOTSZ = TOTSZ + VECSZ
      IF (TOTSZ .GT. SKPSZ) GO TO 40
      CALL PLOT (X2,Y2,2)
      X1 = X2
      Y1 = Y2
      RETURN
C
C     *** ITS TIME FOR A LABEL -- LOCATE START
   40 RATIO = (VECSZ - TOTSZ + SKPSZ)/VECSZ
      X1L = X1 + RATIO*(X2-X1)
      Y1L = Y1 + RATIO*(Y2-Y1)
C
C     *** PLOT SUBVECTOR AND REMEMBER THE END POINT
      CALL PLOT (X1L,Y1L,2)
      X1 = X1L
      Y1 = Y1L
C
C     *** RESET ARCLENGTH OF PREEMPTED SEGMENTS
      ARCL = 0.
C
C     *** FIND OUT IF THERE IS ENOUGH ROOM LEFT IN THIS VECTOR
C         FOR THE LABEL
   50 HAVSZ = SQRT ((X2-X1L)**2 + (Y2-Y1L)**2)
      IF (HAVSZ .GE. SZLBL) GO TO 80
C
```

```
C     *** NOT ENOUGH ROOM -- ACCUMULATE ARCLENGTH
      ARCL = ARCL + SQRT ((X2-X1)**2 + (Y2-Y1)**2)
C
C     *** CHECK FOR OVERFLOW OR TOO WIGGLY
      IF (NSV .LT. MXSV  .AND.  ARCL .LT. 1.1*SZLBL) GO TO 70
C
C     *** LINE MUST BE TOO WIGGLY FOR LABEL -- FLUSH IT
      DO 60 I=1,NSV
          CALL PLOT (XSV(I),YSV(I),2)
   60 CONTINUE
C
C     *** RESET FOR LABEL AND START OVER
      NSV = 0
      TOTSZ = SKPSZ
      GO TO 20
C
C     *** SAVE THIS NODE AND RETURN
   70 NSV = NSV+1
      XSV(NSV) = X2
      YSV(NSV) = Y2
      X1 = X2
      Y1 = Y2
      RETURN
C
C     *** CALCULATE THE END OF THE LABEL
C         IM SURE THERE IS AN EASIER WAY TO DO THIS, BUT IT
C         ESCAPES ME
   80 A = (X2-X1)**2 + (Y2-Y1)**2
      B = -2*((X1L-X1)*(X2-X1) + (Y1L-Y1)*(Y2-Y1))
      C = (X1L-X1)**2 + (Y1L-Y1)**2 - SZLBL*SZLBL
C
      SQRTD = SQRT(B*B - 4*A*C)
      T1 = (-B + SQRTD)/(2*A)
      T2 = (-B - SQRTD)/(2*A)
C
C     *** PICK THE MINIMUM T BETWEEN 0-1 (MUST BE ONE)
      IF (T1 .LT. 0.) T1 = 1.
      IF (T2 .LT. 0.) T2 = 1.
      RATIO = T1
      IF (T2 .LT. T1) RATIO = T2
C
C     *** SET LABEL END POINT
      X2L = X1 + RATIO*(X2-X1)
      Y2L = Y1 + RATIO*(Y2-Y1)
C
C     *** CALCULATE LABEL ANGLE
      DX = X2L - X1L
      DY = Y2L - Y1L
      ANG = ATAN2(DY,DX)
C
```

```
      XL = X1L
      YL = Y1L
      COSA = COS(ANG)
      SINA = SIN(ANG)
C
C     *** REVERSE EVERYTHING IF ANGLE IN QUADRANTS 2 OR 3
      IF (DX .GE. 0.) GO TO 90
      XL = X2L
      YL = Y2L
      COSA = -COSA
      SINA = -SINA
      IF (DY .GE. 0.) ANG = ANG-3.1415926536
      IF (DY .LT. 0.) ANG = ANG+3.1415926536
   90 ANGD = ANG*180./3.1415926536
C
C     *** LOCATE AND PLOT LABEL
      XL = XL + OFF*COSA + OFF*SINA
      YL = YL + OFF*SINA - OFF*COSA
C
C     *** CALL NUMBER TO PLOT THE LABEL
      IDG=NCHR-2
      CALL NUMBER (XL,YL,CHRSZ,OZL,ANGD,IDG)
C
C     *** FINISH OFF THIS SEGMENT BY MEANS OF A PSUEDO REENTRY
      TOTSZ = 0.
      SKPSZ = DIST
      NSV = 0
      X1 = X2L
      Y1 = Y2L
      CALL PLOT (X1,Y1,3)
      GO TO 30
C
C*****************************************************************
C                  ENTRY FOR IPEN = 1
C*****************************************************************
C
C     *** PLOT THE SAVED VECTORS IF ANY
  100 IF (NSV .EQ. 0) RETURN
      DO 110 I=1,NSV
          CALL PLOT (XSV(I),YSV(I),2)
  110 CONTINUE
      NSV = 0
      RETURN
C
      END
      SUBROUTINE VVKRV1
      SAVE
C
C    THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C    COMPUTE AN SPLINE UNDER TENSION PASSING THROUGH A SEQUENCE
C    OF PAIRS (X(1),Y(1),....,X(N),Y(N)) IN THE PLANE. THE
C    SLOPES AT THE TWO ENDS OF THE CURVE MAY BE SPECIFIED OR
C    OMITTED. FOR ACTUAL COMPUTATION OF POINTS ON THE CURVE IT
C    IS NECESSARY TO CALL THE SUBROUTINE VVKRV2.
C
C     *** AK CLINE, COMM. ACM 17,4(APR.1974), 221
C         MODIFIED BY GW LUNDBERG/SAI  APR 79
C
C    COMMON VARIABLES ON INPUT --
C    NPTS = THE NUMBER OF POINTS TO BE INTERPOLATED (N.GE.2),
C    X = AN ARRAY CONTAINING THE N X-COORDINATES OF THE
C        POINTS,
C    Y = AN ARRAY CONTAIING THE N Y-COODINATES OF THE
C        POINTS,
C    NSLP = A FLAG FOR ENDPOINT SLOPES.  IF < 0, THIS IS A CLOSED
C        LOOP AND NO SLOPES ARE GIVEN.  IF = 0, THIS IS AN OPEN CURVE
C        AND NO SLOPES ARE GIVEN.  IF = 1, THE FIRST SLOPE IS SUPPLIED.
C        IF =2 , THE SECOND SLOPE IS SUPPLIED.  IF = 3, BOTH
C        SLOPES ARE SUPPLIED.
C    SLP1,SLPN = THE DESIRED VALUES FOR THE SLOPE
C        OF THE CURVE AT (X(1),Y(1)) AND (X(N),Y(N)), RESPEC-
C        TIVELY.  THESE QUANTITIES ARE IN RADIANS AND MEASURED
C        COUNTER CLOCKWISE FROM THE POSITIVE X-AXIS.  THE POSITIVE
C        SENSE OF THE CURVE IS ASSUMED TO BE THAT MOVING FROM THE
C        POINT 1 TO POINT N.
C    XP,YP = ARRAYS OF LENGTH AT LEAST N,
C    SIGMA = THE TENSION FACTOR.  THIS IS NON-ZERO AND
C        INDICATES THE CURVINESS DESIRED.  IF SIGMA IS VERY
C        LARGE (E.G. 50.) THE RESULTING CURVE IS VERY NEARLY A
C        POLYGONAL LINE.  A STANDARD VALUE FOR SIGMA IS 1.
C
C    ON OUTPUT -
C    NPTS,X,Y,SLP1,SLPN, AND SIGMA ARE UNALTERED,
C    XP,YP CONTAIN INFORMATION ABOUT THE CURVATURE OF THE
C        CURVE AT THE GIVEN NODE,
C    S = THE POLYGONAL ARCLENGTH OF THE CURVE.
C
C
      COMMON /VVKRV/ NPTS,NSLP,SLP1,SLPN,LSLP,TANGNT,S,
     1     X(32),Y(32),XP(32),YP(32)
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
      COMMON /EXPVAL/ EXPMAX
      DIMENSION TEMP(32)
C
```

```
      TEMAX=-9999.
      N = NPTS
      NM1 = N-1
      NP1 = N+1
      DELX1 = X(2)-X(1)
      DELY1 = Y(2)-Y(1)
      DELS1 = SQRT(DELX1*DELX1+DELY1*DELY1)
      DX1 = DELX1/DELS1
      DY1 = DELY1/DELS1
C
C      *** DETERMINE SLOPES IF NECESSARY
C         CHECK IF A LOOP
      IF (NSLP .LT. 0) GO TO 80
      JMP = NSLP + 1
C
C     *** SPECIAL HANDLING FOR TWO POINT LINES
      IF (N .EQ. 2) GO TO 40
   10 GO TO (20,30,20,90),JMP
C
C     *** USE SECOND ORDER INTERPOLATION FOR ENDPOINT SLOPES
   20 DELS2 = SQRT((X(3)-X(2))**2+(Y(3)-Y(2))**2)
      DELS12 = DELS1+DELS2
      C1 = -(DELS12+DELS1)/DELS12/DELS1
      C2 = DELS12/DELS1/DELS2
      C3 = -DELS1/DELS12/DELS2
      SX = C1*X(1)+C2*X(2)+C3*X(3)
      SY = C1*Y(1)+C2*Y(2)+C3*Y(3)
      SLP1 = ATAN2(SY,SX)
      IF (NSLP .EQ. 2) GO TO 90
C
   30 DELNM1 = SQRT((X(N-2)-X(NM1))**2+(Y(N-2)-Y(NM1))**2)
      DELN = SQRT((X(NM1)-X(N))**2+(Y(NM1)-Y(N))**2)
      DELNN = DELNM1+DELN
      C1 = (DELNN+DELN)/DELNN/DELN
      C2 = -DELNN/DELN/DELNM1
      C3 = DELN/DELNN/DELNM1
      SX = C3*X(N-2)+C2*X(NM1)+C1*X(N)
      SY = C3*Y(N-2)+C2*Y(NM1)+C1*Y(N)
      SLPN = ATAN2(SY,SX)
      GO TO 90
C
C     *** ONLY TWO POINTS SUPPLIED
   40 GO TO (50,60,70,90),JMP
C
C     *** NO SLOPES GIVEN -- USE STRAIGHT LINE
C         (ACTUALLY, THIS CASE SHOULD NEVER OCCUR)
   50 XP(1) = 0.
      XP(2) = 0.
      YP(1) = 0.
      YP(2) = 0.
      S = SQRT ((X(2)-X(1))**2 + (Y(2)-Y(1))**2)
      RETURN
C
C     *** FIRST SLOPE GIVEN
   60 SLPN = ATAN2 (Y(2)-Y(1)-SLP1*(X(2)-X(1)),
     1              X(2)-X(1)-SLP1*(Y(2)-Y(1)))
      GO TO 90
C
C     *** LAST SLOPE GIVEN
   70 SLP1 = ATAN2 (Y(2)-Y(1)-SLPN*(X(2)-X(1)),
     1              X(2)-X(1)-SLPN*(Y(2)-Y(1)))
      GO TO 90
C
C     *** CLOSED LOOP -- PERIODIC SPLINE -- CALCULATE SLOPES
C         FOR JOIN
   80 DELN = SQRT((X(NM1)-X(N))**2+(Y(NM1)-Y(N))**2)
      DELNN = DELS1+DELN
      C1 = -DELS1/DELN/DELNN
      C2 = (DELS1-DELN)/DELS1/DELN
      C3 = DELN/DELNN/DELS1
      SX = C1*X(NM1)+C2*X(1)+C3*X(2)
      SY = C1*Y(NM1)+C2*Y(1)+C3*Y(2)
      IF (SX.EQ.O. .AND. SY.EQ.O.) SX = 1
      SLP1 = ATAN2(SY,SX)
      SLPN = SLP1
C
C     *** SET BOTH SLOPES
   90 SLPP1 = SLP1
      SLPPN = SLPN
C
C     *** SET UP RIGHT HAND SIDES OF TRIDIAGONAL LINEAR SYSTEM FOR
C         XP AND YP
  100 XP(1) = DX1-COS(SLPP1)
      YP(1) = DY1-SIN(SLPP1)
      TEMP(1) = DELS1
      S = DELS1
      IF (N.EQ.2) GO TO 120
      DO 110 I=2,NM1
          DELX2 = X(I+1)-X(I)
          DELY2 = Y(I+1)-Y(I)
          DELS2 = SQRT(DELX2*DELX2+DELY2*DELY2)
          DX2 = DELX2/DELS2
          DY2 = DELY2/DELS2
          XP(I) = DX2-DX1
          YP(I) = DY2-DY1
          TEMP(I) = DELS2
          TEMAX = AMAX1(TEMAX,TEMP(I))
          DELX1 = DELX2
          DELY1 = DELY2
          DELS1 = DELS2
          DX1 = DX2
          DY1 = DY2
C
C     *** ACCUMULATE POLYGONAL ARCLENGTH
          S =S+DELS1
  110 CONTINUE
  120 XP(N) = COS(SLPPN)-DX1
      YP(N) = SIN(SLPPN)-DY1
C
C     *** DENORMALIZE TENSION FACTOR
      SIGMAP = ABS(SIGMA)*FLOAT(N-1)/S
      DELT1=SIGMAP*TEMAX
      IF (DELT1.LT.EXPMAX) GO TO 130
      SIGMAP=0.9*EXPMAX/TEMAX
      SGN=1.0
      IF (SIGMAP.LT.0.) SGN=-1.0
      SIGMA=SIGMAP*SGN*S/FLOAT (N-1)
  130 CONTINUE
C
C     *** PERFORM FORWARD ELIMINATION ON TRIDIAGONAL SYSTEM
      DELS = SIGMAP*TEMP(1)
      EXPS = EXP(DELS)
      SINHS = .5*(EXPS-1./EXPS)
      SINHIN = 1./(TEMP(1)*SINHS)
      DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHS)
      DIAGIN = 1./DIAG1
      XP(1) = DIAGIN*XP(1)
      YP(1) = DIAGIN*YP(1)
      SPDIAG = SINHIN*(SINHS-DELS)
      TEMP(1) = DIAGIN*SPDIAG
      IF (N.EQ.2) GO TO 150
      DO 140 I=2,NM1
          DELS = SIGMAP*TEMP(I)
          EXPS = EXP(DELS)
          SINHS = .5*(EXPS-1./EXPS)
          SINHIN = 1./(TEMP(I)*SINHS)
          DIAG2 = SINHIN*(DELS*(.5*(EXPS+1./EXPS))-SINHS)
          DIAGIN = 1./(DIAG1+DIAG2-SPDIAG*TEMP(I-1))
          XP(I) = DIAGIN*(XP(I)-SPDIAG*XP(I-1))
          YP(I) = DIAGIN*(YP(I)-SPDIAG*YP(I-1))
          SPDIAG = SINHIN*(SINHS-DELS)
          TEMP(I) = DIAGIN*SPDIAG
          DIAG1 = DIAG2
  140 CONTINUE
  150 DIAGIN = 1./(DIAG1-SPDIAG*TEMP(NM1))
      XP(N) = DIAGIN*(XP(N)-SPDIAG*XP(NM1))
      YP(N) = DIAGIN*(YP(N)-SPDIAG*YP(NM1))
C
C     *** PERFORM BACK SUBSTITUTION
      DO 160 I=2,N
```

```
          IBAK = NP1-I                                                 AY  201
          XP(IBAK) = XP(IBAK)-TEMP(IBAK)*XP(IBAK+1)                    AY  202
          YP(IBAK) = YP(IBAK)-TEMP(IBAK)*YP(IBAK+1)                    AY  203
  160 CONTINUE                                                         AY  204
      RETURN                                                           AY  205
      END                                                              AY  206-
      SUBROUTINE VVKRV2 (T,XS,YS)
      SAVE
C
C     THIS SUBROUTINE PERFORMS THE MAPPING OF POINTS IN THE
C     INTERVAL (0.,1.) ONTO A CURVE IN THE PLANE.  THE SUBROUTINE
C     VVKRV1 SHOULD BE CALLED EARLIER TO DETERMINE CERTAIN
C     NECESSARY PARAMETERS.  THE RESULTING CURVE HAS A PARAMETRIC
C     REPRESENTATION BOTH OF WHOSE COMPONENTS ARE SPLINES UNDER
C     TENSION AND FUNCTIONS OF THE POLYGONAL ARCLENGTH PARAMETER.
C
C     COMMON VARIABLES AND PARAMETERS ON INPUT--
C     T = A REAL VALUE OF ABSOLUTE VALUE LESS THAN OR
C         EQUAL TO 1.  TO BE MAPPED TO A POINT ON THE CURVE.  THE
C         SIGN OF T IS IGNORED AND THE INTERVAL (0.,1.) IS MAPPED
C         ONTO THE ENTIRE CURVE.  IF T IS NEGATIVE THIS INDICATES
C         THAT THE SUBROUTINE HAS BEEN CALLED PREVIOUSLY (WITH ALL
C         OTHER INPUT VARIABLES UNALTERED) AND THAT THIS VALUE OF
C         T EXCEEDS THE PREVIOUS VALUE IN ABSOLUTE VALUE.  WITH
C         SUCH INFORMATION THE SUBROUTINE IS ABLE TO MAP THE POINT
C         MUCH MORE RAPIDLY.  THUS IF THE USER SEEKS TO MAP A
C         SEQUENCE OF POINTS ONTO THE SAME CURVE, EFFICIENCY IS
C         GAINED BY ORDERING THE VALUES INCREASING IN MAGNITUDE
C         AND SETTING THE SIGNS OF ALL BUT THE FIRST, NEGATIVE,
C     NPTS = THE NUMBER OF POINTS WHICH WERE INTERPOLATED
C         TO DETERMINE THE CURVE,
C     X,Y = ARRAYS CONTAINING THE X- AND Y-COORDINATES
C         OF THE INTERPOLATED POINTS,
C     XP,YP = THE ARRAYS OUTPUT FROM VVKRV2 CONTAINING
C         CURVATURE INFORMATION,
C     S = THE POLYGONAL ARCLENGTH OF THE CURVE,
C     SIGMA = THE TENSION FACTOR (ITS SIGN IS IGNORED).
C     LSLP = A FLAG WHICH IF >0 REQUESTS THE SLOPE
C     TANGNT = THE SLOPE REQUESTED BY LSLP
C
C     THE PARAMETERS NPTS,X,Y,XP,YP,S, AND SIGMA SHOULD BE INPUT
C     UNALTERED FROM THE OUTPUT OF VVKRV1.
C
C     ON  OUTPUT--
C     XS,YS = THE X- AND Y-COORDINATES OF THE IMAGE
C         POINT ON THE CURVE.
C     T,NPTS,X,Y,XP,YP,S, AND SIGMA ARE UNALTERED.
C
      COMMON /VVKRV/ NPTS,NSLP,SLP1,SLPN,LSLP,TANGNT,S,
     1     X(32),Y(32),XP(32),YP(32)
      COMMON /CNTRL/ SIG,SIGMA,INFO,NPTO,TSRT,DTIM,Z1,Z2,DCON,EHC,EXN,FL
     1ST,TLST
C
      N = NPTS
C
C     *** DENORMALIZE SIGMA
```

```
      SIGMAP = ABS(SIGMA)*FLOAT(N-1)/S
C
C     *** STRETCH UNIT INTERVAL INTO ARCLENGTH DISTANCE
      TN = ABS(T*S)
C
C     *** FOR NEGATIVE T START SEARCH WHERE PREVIOUSLY TERMINATED.
C         OTHERWISE START FROM BEGINNING
      IF (T.LT.0.) GO TO 10
      I1 = 2
      XS = X(1)
      YS = Y(1)
      SUM = 0.
      IF (T.LT.0.) RETURN
C
C     *** DETERMINE INTO WHICH SEGMENT TN IS MAPPED
   10 DO 30 I=I1,N
          DELX = X(I)-X(I-1)
          DELY = Y(I)-Y(I-1)
          DELS = SQRT(DELX*DELX+DELY*DELY)
          IF (SUM+DELS-TN) 20,40,40
   20     SUM = SUM+DELS
   30 CONTINUE
C
C     *** IF ABS(T) IS GREATER THAN 1., RETURN TERMINAL POINT ON
C         CURVE
C
      XS = X(N)
      YS = Y(N)
      RETURN
C
C     *** SET UP AND PERFORM INTERPOLATION
   40 DEL1 = TN-SUM
      DEL2 = DELS-DEL1
      EXPS1 = EXP(SIGMAP*DEL1)
      SINHD1 = .5*(EXPS1-1./EXPS1)
      EXPS2 = EXP(SIGMAP*DEL2)
      SINHD2 = .5*(EXPS2-1./EXPS2)
      EXPS = EXPS1*EXPS2
      SINHS = .5*(EXPS-1./EXPS)
      XS = (XP(I)*SINHD1+XP(I-1)*SINHD2)/SINHS+
     1     ((X(I)-XP(I))*DEL1+(X(I-1)-XP(I-1))*DEL2)/DELS
      YS = (YP(I)*SINHD1+YP(I-1)*SINHD2)/SINHS+
     1     ((Y(I)-YP(I))*DEL1+(Y(I-1)-YP(I-1))*DEL2)/DELS
      I1 = I
      IF (LSLP .EQ. 0) RETURN
C
C     *** CALCULATE SLOPE
      COSHD1 = 0.5*(EXPS1+1.0/EXPS1)*SIGMAP
      COSHD2 = 0.5*(EXPS2+1.0/EXPS2)*SIGMAP
      XT = (XP(I)*COSHD1-XP(I-1)*COSHD2)/SINHS+
1      ((X(I)-XP(I))-(X(I-1)*XP(I-1)))/DELS                     AZ  101
 YT = (YP(I)*COSHD1-YP(I-1)*COSHD2)/SINHS+                      AZ  102
1      ((Y(I)-YP(I))-(Y(I-1)-YP(I-1)))/DELS                     AZ  103
 TANGNT = ATAN2 (YT,XT)                                         AZ  104
 RETURN                                                         AZ  105
 END                                                            AZ  106-
```

```
      SUBROUTINE AXES (X,Y,FIRSTV,FINALV,SCALE,TSTEP,ASTEP,NDEC,ANGLE,IBBA
     1CD,NCHAR,LABEL)                                                   BA
C                                                                       BA
C     *** SAI SUBROUTINE AXES                                           BA
C         GW LUNDBERG/SAI  DEC 76                                       BA
C                                                                       BA
C         X,Y = COORDINATES IN INCHES OF AXIS LINE STARTING             BA
C               POINT                                                   BA
C         FIRSTV = STARTING VALUE FOR THE AXIS                          BA
C         FINALV = ENDING VALUE FOR THE AXIS                            BA
C         SCALE = INCHES/UNIT FOR FIRSTV,FINALV,TSTEP,ASTEP             BA
C         TSTEP = STEP SIZE FOR TICS                                    BA
C         ASTEP = STEP SIZE FOR LABELED TICS                            BA
C         NDEC = FORMAT FOR LABELS -- SEE SUBROUTINE NUMBER             BA
C         ANGLE = ANGLE OF AXIS IN DEGRES FROM HORIZONTAL               BA
C         IBCD = THE AXIS TITLE AS ARRAY OR HOLLERITH STRING            BA
C         NCHAR = NUMBER OF CHARACTERS IN TITLE                         BA
C               ) 0, TIC MARKS, ANNOTATION AND TITLE PLOTTED ON         BA
C                   CLOCKWISE SIDE OF AXIS LINE                         BA
C               ( 0, ON COUNTER CLOCKWISE SIDE                          BA
C                                                                       BA
C         THIS ROUTINE WAS WRITTEN FOR A MATRIX PLOTTER -- IT DOES      BA
C         NOT OPTIMIZE PEN MOVEMENTS.  THE ROUTINE SHOULD BE MACHINE    BA
C         INDEPENDENT                                                   BA
C                                                                       BA
C                                                                       BA
C     *** FOLLOWING ARE ADJUSTABLE -- IF LABEL ) 0, ALL TICS ARE        BA
C         LABELED, IF LABEL = 0, THE LAST TIC IS NOT LABELED,           BA
C         IF LABEL ( 0, THE FIRST AND LAST ARE NOT LABELED              BA
C                                                                       BA
C     *** DEFINE VARIOUS CHARACTER SIZES                                BA
      SAVE                                                              BA
      COMMON /PLTVEC/ HCT(20),OT(20),NH,OHC,HCG,PLTGRD,OXN,XNG,HC1,XN1,TBA
     1ICSIZ,DIGSIZ,CHRSIZ                                               BA
      INTEGER IBCD(3)                                                   BA
C                                                                       BA
C     *** STEP 1 -- DRAW AXIS AND TIC MARKS                             BA
C         ---- -    ---- ---- --- --- -----                             BA
C                                                                       BA
C     *** MOVE PEN TO START OF AXIS                                     BA
      X0=X                                                              BA
      Y0=Y                                                              BA
      CALL PLOT (X0,Y0,3)                                               BA
C                                                                       BA
C     *** LOCATE THE OTHER END AND DRAW AXIS                            BA
      COSA=COS(ANGLE*0.017453294)                                       BA
      SINA=SIN(ANGLE*0.017453294)                                       BA
      AXLEN=(FINALV-FIRSTV)*SCALE                                       BA
      X1=X0+AXLEN*COSA                                                  BA
      Y1=Y0+AXLEN*SINA                                                  BA
      CALL PLOT (X1,Y1,2)
C
C     *** ADD THE TIC MARKS ON WRONG SIDE OF AXIS
      POS=FLOAT(ISIGN(1,NCHAR))
      IF (TSTEP.EQ.0.) GO TO 20
      NTIC=(FINALV-FIRSTV)/TSTEP+1.5
      DO 10 J=1,NTIC
C
C         *** MOVE PEN TO START OF TIC
      X1=X0+FLOAT(J-1)*TSTEP*SCALE*COSA
      Y1=Y0+FLOAT(J-1)*TSTEP*SCALE*SINA
      CALL PLOT (X1,Y1,3)
C
C         *** DRAW A TIC NORMAL TO AXIS
      X1=X1+TICSIZ*POS*SINA
      Y1=Y1-TICSIZ*POS*COSA
      CALL PLOT (X1,Y1,2)
   10 CONTINUE
C
C     *** STEP 2 -- SET IN LABELED TICS ON CORRECT SIDE OF AXIS
C         ---- -    --- -- ------- ---- -- ------- ---- -- ----
C
   20 IF (ASTEP.EQ.0.) GO TO 50
      NTIC=(FINALV-FIRSTV)/ASTEP+1.5
      NFRST=1
      NLST=NTIC
      IF (LABEL.LE.0) NLST=NLST-1
      IF (LABEL.LT.0) NFRST=NFRST+1
      DO 30 J=NFRST,NLST
      X1=X0+FLOAT(J-1)*ASTEP*SCALE*COSA
      Y1=Y0+FLOAT(J-1)*ASTEP*SCALE*SINA
      CALL PLOT (X1,Y1,3)
C
      X1=X1-TICSIZ*POS*SINA
      Y1=Y1+TICSIZ*POS*COSA
      CALL PLOT (X1,Y1,2)
   30 CONTINUE
C
C     *** STEP 3 -- ANNOTATE THE TIC MARKS
C         ---- -    -------- --- --- -----
C
C     *** DETERMINE PERPENDICULAR OFFSET TO BOTTOM OF CHARACTER
      OFFSET=TICSIZ+0.03
      IF (POS.NE.1.0) OFFSET=OFFSET+DIGSIZ
C
C     *** CALCULATE LOCATION OF FIRST CHARACTER
      X0=X0-DIGSIZ*COSA
      Y0=Y0-DIGSIZ*SINA
      X0=X0-OFFSET*POS*SINA
      Y0=Y0+OFFSET*POS*COSA
```

```
C
C     *** ANNOTATE THE TIC MARKS
      NFRST=1
      NLAST=NTIC
      IF (LABEL.LE.0) NLAST=NLAST-1
      IF (LABEL.LT.0) NFRST=NFRST+1
      DO 40 J=NFRST,NLAST
C
C         *** GET FLOATING POINT VALUE OF ANNOTATION
      FPN=FIRSTV+FLOAT(J-1)*ASTEP
C
C         *** GET LOCATION AND PLOT FPN
      X1=X0+FLOAT(J-1)*ASTEP*SCALE*COSA
      Y1=Y0+FLOAT(J-1)*ASTEP*SCALE*SINA
      CALL NUMBER (X1,Y1,DIGSIZ,FPN,ANGLE,NDEC)
   40 CONTINUE
C
C     *** STEP 4 -- ADD AXIS TITLE
C         ---- -    --- ---- -----
C
   50 IF (NCHAR.EQ.0) GO TO 60
C     *** SET TITLE OFFSET -- DISTANCE FROM AXIS TO CHARACTERS
      OFFSET=TICSIZ+DIGSIZ+0.10
      IF (NCHAR.LT.0) OFFSET=OFFSET+CHRSIZ
C
C     *** CALCULATE TITLE SIZE
      TSIZ=CHRSIZ*FLOAT(IABS(NCHAR))
C
C     *** CALCULATE OFFSET FROM BEGINNING OF AXIS TO FIRST CHARACTER
      OFF=0.5*(AXLEN-TSIZ)
C
C     *** CALCULATE LOCATION OF FIRST CHARACTER AND PLOT TITLE
      X0=X+OFF*COSA
      Y0=Y+OFF*SINA
      X0=X0-OFFSET*POS*SINA
      Y0=Y0+OFFSET*POS*COSA
      NCCAR=IABS(NCHAR)
      CALL SYMBOL (X0,Y0,CHRSIZ,IBCD,ANGLE,NCCAR)
C
C         FINISHED
   60 RETURN
      END
      SUBROUTINE FRAME (X,Y,FRSTX,FINX,SIZX,TICX,STEPX,NDECX,FRSTY,FINY,
     1SIZY,TICY,STEPY,NDECY,LBLBOT,NB,LBLLFT,NL,LBLTOP,NT,LBLRGT,NR)
C
C     *** PURPOSE -- PREPARES AN ANNOTATED FOUR SIDED FRAME
C         WITH LOWER LEFT CORNER AT (X,Y)
C             GWL/SAI  MARCH  77
C
C         SEE SUBROUTINE AXES FOR DESCRIPTION OF ARGUMENTS
C
      SAVE
      COMMON /PLTVEC/ HCT(20),OT(20),NH,OHC,HCG,PLTGRD,OXN,XNG,HCC,XNC,T
     1ICZ,DIGZ,CHRZ
      INTEGER LBLBOT(3),LBLTOP(3),LBLLFT(3),LBLRGT(3)
C
C     *** CALCULATE THE AXES LENGTHS
      XLEN=(FINX-FRSTX)*SIZX
      YLEN=(FINY-FRSTY)*SIZY
C
C     *** PLACE GRIDDED LINES ON PLOT
C
      IF (ABS(PLTGRD).EQ.0.) GO TO 80
      IF (PLTGRD.LT.0.) GO TO 50
      CALL NEWPEN (2)
      STINC=(STEPY/10.)*SIZY
      ILP=FINY/STEPY+.005
      ILP=ILP*10-1
      STVAL=STINC
      J=1
      DO 20 I=1,ILP
      IF (J.EQ.2) GO TO 10
      CALL PLOT (X,STVAL,3)
      CALL PLOT (XLEN,STVAL,2)
      STVAL=STVAL+STINC
      J=2
      GO TO 20
   10 CALL PLOT (XLEN,STVAL,3)
      CALL PLOT (X,STVAL,2)
      STVAL=STVAL+STINC
      J=1
   20 CONTINUE
      STINC=(STEPX/10.)*SIZX
      ILP=FINX/STEPX+.005
      ILP=ILP*10-1
      STVAL=STINC
      J=1
      DO 40 I=1,ILP
      IF (J.EQ.2) GO TO 30
      CALL PLOT (STVAL,Y,3)
      CALL PLOT (STVAL,YLEN,2)
      STVAL=STVAL+STINC
      J=2
      GO TO 40
   30 CALL PLOT (STVAL,YLEN,3)
      CALL PLOT (STVAL,Y,2)
      STVAL=STVAL+STINC
      J=1
   40 CONTINUE
      CALL NEWPEN (1)
      GO TO 80
C
C  PLOT GRIDDED LINES WITH MATRIX PLOTTER
C
   50 IPLT=IFIX(PLTGRD)
      CALL NEWPEN (IPLT)
      STINC=(STEPY/10.)*SIZY
      ILP=IFIX(FINY/STEPY+0.005)*10-1
      STVAL=STINC
      DO 60 I=1,ILP
      IF (MOD(I,10).EQ.0) CALL NEWPEN (IPLT+1)
      CALL PLOT (X+0.01,STVAL,3)
      CALL PLOT (XLEN,STVAL,2)
      IF (MOD(I,10).EQ.0) CALL NEWPEN (IPLT)
      STVAL=STVAL+STINC
   60 CONTINUE
      STINC=(STEPX/10.)*SIZX
      ILP=IFIX(FINX/STEPX+0.005)*10-1
      STVAL=STINC
      DO 70 I=1,ILP
      IF (MOD(I,10).EQ.0) CALL NEWPEN (IPLT+1)
      CALL PLOT (STVAL,Y-0.02,3)
      CALL PLOT (STVAL,Y,3)
      CALL PLOT (STVAL,YLEN,2)
      IF (MOD(I,10).EQ.0) CALL NEWPEN (IPLT)
      STVAL=STVAL+STINC
   70 CONTINUE
      CALL NEWPEN (0)
   80 CONTINUE
C
C     *** PLOT THE FOUR SIDES WITH ANNOTATIONS
      CALL AXES (X,Y,FRSTX,FINX,SIZX,TICX,STEPX,NDECX,0.,LBLBOT,-NB,1)
      CALL AXES (X,Y,FRSTY,FINY,SIZY,TICY,STEPY,NDECY,90.,LBLLFT,NL,1)
      CALL AXES (X,Y+YLEN,FRSTX,FINX,SIZX,TICX,STEPX,NDECX,0.,LBLTOP,NT,
     1-1)
      CALL AXES (X+XLEN,Y,FRSTY,FINY,SIZY,TICY,STEPY,NDECY,90.,LBLRGT,-N
     1R,-1)
C
      RETURN
      END
```